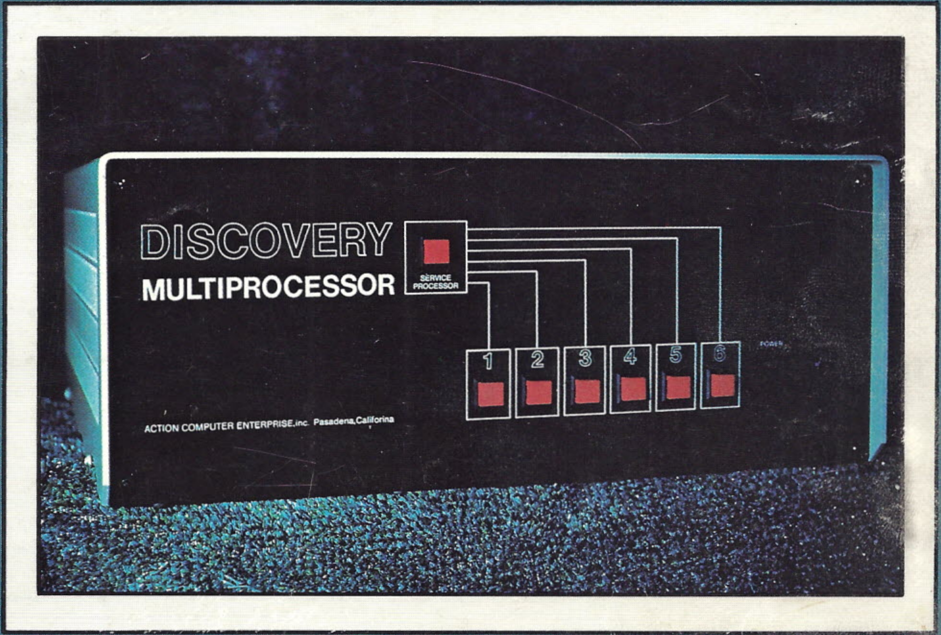
kilobaud
June 1981
USA $2.95/DM 9.80

MICROCOMPUTING T.M.

DESKTOP COMPUTING™

Multiprocessor VS Multitask

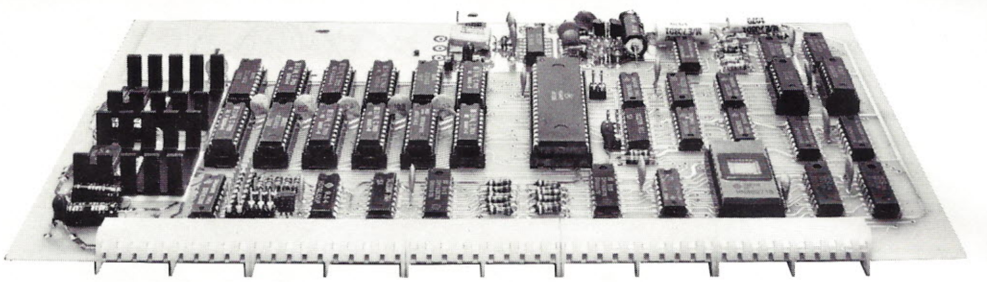


Systems Square Off In Business Arena

On Easy Street with the Apple II □ The S-50 Bus Strikes Back! □ Teach a Sorcerer New Tricks □ OSI Baud Mod □ Write a Happy Ending with PET □ North Star Quiz □ Struggle for Survival Simulation □ All About Data Structures □ Light-Fingered Computing □ Be a Better Writer with SWTP

Now! Color for Your . . .

SYSTEM-50



Introducing COLORAMA-50™ Percom's SS-50 Bus Color VDG

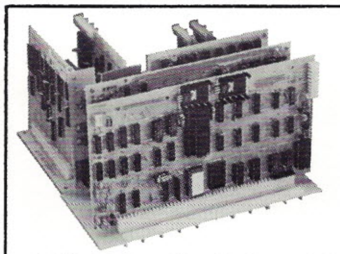
Introductory Price $219.95

Featuring . . .

- **Eleven display formats** including 8-color semigraphics, 4-color graphics, 2-color high density graphics and 2-color alphanumerics.
  Moreover, two- and four-color displays may be switched between primary and complementary color sets under software control or from the keyboard.
  Full graphic resolutions range from 64 x 64 picture elements to 256 x 192 picture elements.
- **Instant display control:** The COLO*RAMA*-50™ is memory mapped: your MPU has direct, instant access to display RAM and display control registers.
- **Low-cost Modulator Option for Color TV Interface:** The COLO*RAMA*-50™ provides for installation of an inexpensive RF modulator such as Radio Shack PN 277-122 for operation using a color TV.
- **Mix in Sound:** With the optional modulator installed, you can complement your colorful displays with software-controlled audio.
- **Extended Addressing:** The COLO*RAMA*-50™ is compatible with the SS-50A bus *and* the extended-address SS-50C bus. Map the board into any of the sixteen 64-Kbyte banks of the 1-Mbyte SS-50C address space. The COLO*RAMA*-50™ card "defaults" to the first (lowest) bank for the SS-50A bus.
- **Cassettee I/O Option:** Add a few inexpensive components to the on-card circuitry provided and use an audio cassette for program/data storage.
- **Provision for On-Card Firmware:** Put your display operating system, cassette control program, etc. right on the COLO*RAMA*-50™ card in a 2516 (5-volt 2716) EPROM. Resides in the top 2-Kbyte of the card memory space.
- **Operating Software:** Included in the comprehensive users manual is a listing of a display operating system and cassette controller that may be implemented as a callable subroutine function from BASIC or existing operating systems. The programs are optionally available in a plug-in ROM for just $69.95.

System Requirements

The COLO*RAMA*-50™ is pin- and outline-compatible with the Percom System-50™ bus, the SS-50A (SS-50) bus and the SS-50C bus. The composite video-sync signal output will directly drive a color (or BW) video monitor. The output may be modulated for operation with a standard (NTSC) TV set. A modulator is not included. The COLO*RAMA*-50™ card occupies 8-Kbytes of memory in the upper half of a 64-Kbyte memory space. Included on-card is 1-Kbyte of display RAM which will accommodate alphanumeric displays, semigraphic displays and two low-density full-graphic displays. For the higher density graphic displays, additional display RAM is required. The optional RAM ICs may be installed on the card.



SS-50 Bus Department Store

Nobody supports the SS-50 bus like Percom:

- √ SS-50 Bus/Single-Board Computers with I/O ports & memory
- √ Static and Dynamic RAM cards — memory expansion kits
- √ LFD-400/800 1-, 2- and 3-drive mini-disk systems
- √ Color and monochrome memory-mapped display controllers
- √ Extendable 7-slot SS-50 bus motherboards
- √ Versatile prototyping boards: SS-50 and SS-30 bus
- √ Field-proven software: monitors, operating systems, drivers, editors, assemblers, debuggers and HLLs.

For quality Percom SS-50 bus products, see your nearby authorized Percom dealer. To order direct, call **toll-free, 1-800-527-1592.** Prices and specifications subject to change without notice. Prices do not include shipping and handling.

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421

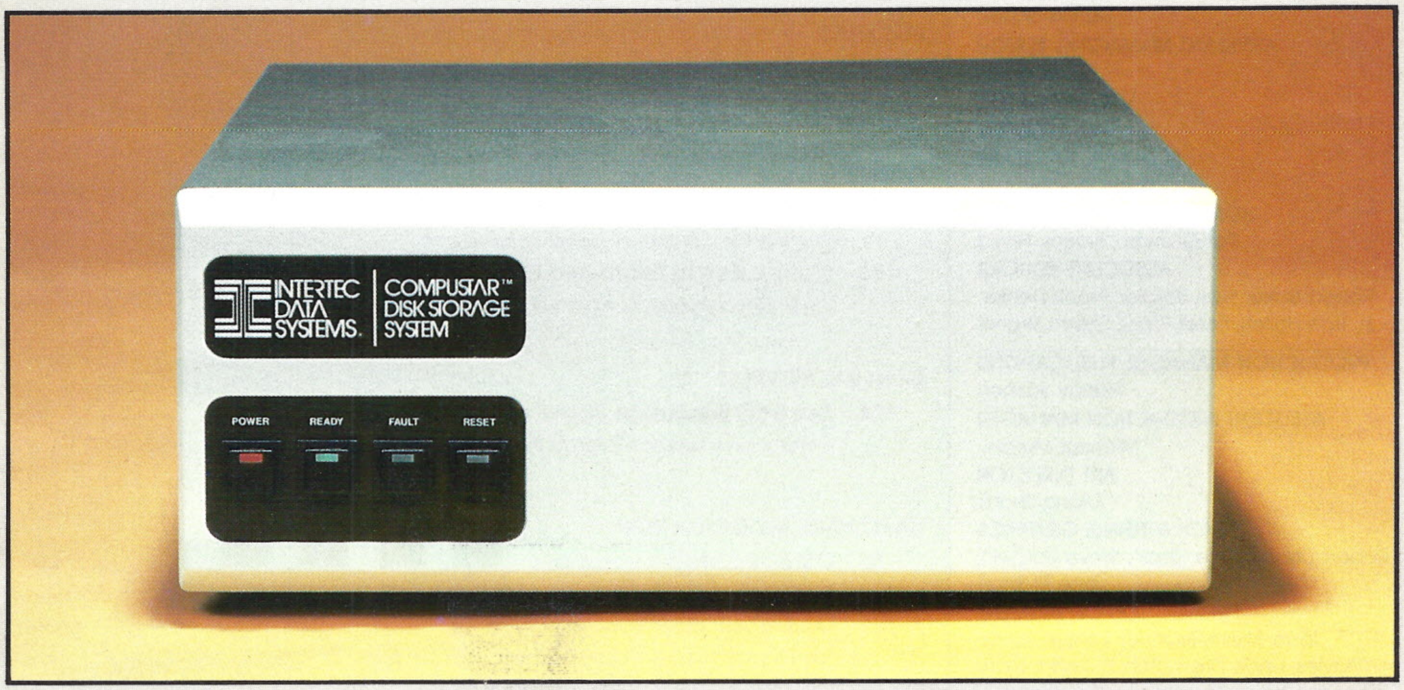
PERCOM

✓1

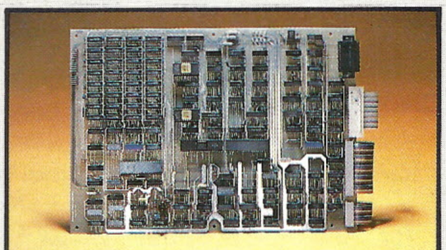
™ trademark of Percom Data Company, Inc.

SOLVE THE STORAGE SHORTAGE

with ten multi-user megabytes



CompuStar's 8-inch Winchester



Intertec's exclusive 255 User Controller

CompuStar Cable Assembly

If you could think of just one way to improve our phenomenally popular SuperBrain, what would it be? More disk storage? Well, we already thought about it. And for only a few thousand dollars for a whopping 10 megabytes of lightning-fast storage, it's nothing short of another major breakthrough! From the company that wrote the book on price/performance . . . Intertec.

Our New CompuStar™ 10 Megabyte Disk Storage System (called a DSS) features an 8 inch Winchester drive packaged in an attractive, compact desktop enclosure. Complete with disk, controller and power supply. Just plug it into the Z80 adaptor of your SuperBrain and turn it on.* It's so quiet, you'll hardly know it's there. But, you'll quickly be astounded with its awesome power and amazing speed.

The secret behind our CompuStar DSS is its unique controller/multiplexor. It allows many terminals to "share" the resources of a single disk. So, not only can you use the DSS with your SuperBrain, you can configure multiple user stations using our new series of CompuStar™ terminals, called Video Processing Units or VPU's™.

Four CompuStar VPU's are available. At prices starting at less than $2,500. Some models are designed to operate as stand-alone microcomputers, with internal disk storage. Just like your SuperBrain. Each model features its own 64K of RAM and can be "daisy-chained" into a powerful multi-user network. Just connect one VPU into the next. Using easy-to-install cable assemblies. Connect up to 255 users in a single system. One at a time. As you need them.

Whether you need an extra 10 megabytes for your SuperBrain or an enormous multi-user network, the CompuStar™ DSS solves your storage shortage problems. Sensibly. And economically. Plus, your investment is protected by a nationwide service network with outlets in most major U.S. cities. Providing efficient on-site or depot maintenance.

Get a demonstration of this extraordinary new system today. Call or write now for the name and address of your nearest CompuStar dealer.

INTERTEC DATA SYSTEMS®

2300 Broad River Rd. Columbia, S.C. 29210
(803) 798-9100 TWX: 810-666-2115

*Some models require hardware/software modification.

MICROCOMPUTING

APPLICATIONS

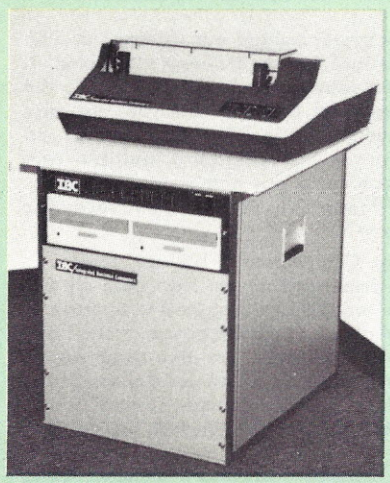
BUSINESS

GAMES

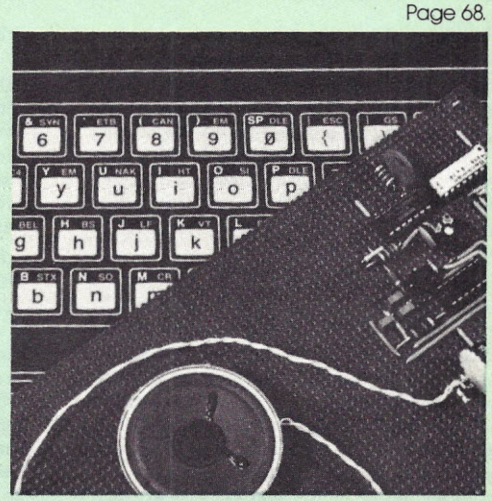
GENERAL INTEREST

HARDWARE MODIFICATION

HARDWARE PROJECTS



Page 34.

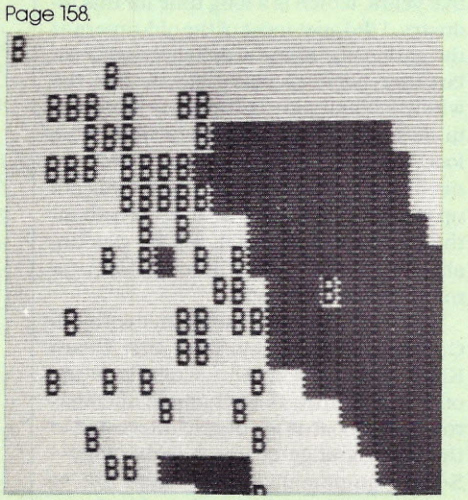


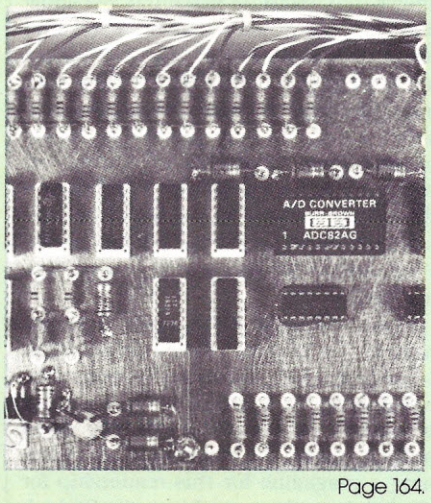
Page 68.

Contents: June 1981

Volume V
No. 6

Page 158.





Page 164.

This month:

As we sit here looking out at a cold, damp May morning, we find it hard to believe that the 1982 editorial calendar is already begging for attention.

But that's the way it is in the publishing business. The successive issues twirl dizzily across the screen of time like in a scene from *The Front Page*, each a memory before it even becomes a reality, each moving from the future to the past with only a brief pit stop in the present.

Our problem is compounded by the ever-changing face of the micro world. By the time we cover a new development, a dozen others pop up.

So the 1982 editorial calendar emerges, paradoxically, both full and unformed. Which of all the many topics do we choose to explore? Education, business, data communications, the home, medicine, science, social science, agriculture—which are most deserving of our attention?

At times like these, only one decisive course of action presents itself—throw the problem into the readers' laps. So here's your chance: What areas of the microworld do *you* want us to cover?

Please respond quickly. 1983 is just around the corner.

—The Editors

Next month:

The July issue will feature computer graphics—how to generate them on your system and how they can be used both in business and in the home. Also, the winning entries of our first annual graphics contest will be displayed.

The cover:

Multi-user systems: The Discovery Multiprocessor, top (courtesy of Action Computer Enterprises), and the System 40, bottom (courtesy of Integrated Business Computers).

Kilobaud *Microcomputing* (ISSN 0192-4575) is published monthly by Wayne Green, Inc., 80 Pine St., Peterborough NH 03458. Subscription rates in U.S. are $25 for one year and $53 for three years. In Canada: $27 for one year only, U.S. funds. Foreign subscriptions (surface mail)—$35 for one year only, U.S. funds. Foreign air mail subscriptions —$62 for one year only, U.S. funds. Canadian Distributor: Micron Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. In Europe, contact: Monika Nedela, Markstr. 3, D-7778 Markdorf, W. Germany. South African Distributor: KB Microcomputing, PO Box 782815, Sandton, South Africa 2146. Australian Distributor: Electronic Concepts, Attention: Rudi Hoess, 55 Clarence Street, Sidney 2000, Australia. Second-class postage paid at Peterborough NH 03458 and at additional mailing offices. Phone: 603-924-3873.

By Wayne Green

Ho-Hum Faire

Well-Attended, But Not Much To See

The Faire

Since most of you missed the sixth annual Computer Faire in San Francisco, let me briefly tell you about it. I did enjoy seeing a lot of friends, and a little bit of new equipment—precious little.

The regular air fare runs close to $1000 for the round trip to the coast. Fortunately, there is a discount airline—U.S. Air—with a price in the $300 range. It's a sort of red-eye special, arriving around midnight at Oakland. I got to bed by 2 AM (5 AM eastern time). Except for the arrival time, the airline is fine, with comfortable seats and the food about the same as most other airlines—not like that confounded Aer Lingus line, which has seats so close together there is no room for your knees. I like to fly to Ireland (we have a plant there now), but it will be via London or Munich from now on.

Since there was a scheduled meeting with the Instant Software reps the day before the Faire, I arrived a day early and spent a good deal of the day talking with our rep organization. That evening Sherry and I managed tickets to Sweeny Todd, a broadway show now resident in San Francisco. Ugh.

The Faire was a replay of the last one I attended. There were supposed to be 500 booths, and I'm sure there were. It took me two full days to get around to all of them even for a short visit. The exhibit halls (two of them) were either busy or packed much of the time. I kept asking everyone I met what they had found new at the show, and the answer was the same each time: the Osborne computer, and little else. The O-1 system drew crowds and elicited comments from many. Many people seemed anxious to get their hands on one. Others were critical of this or that design point.

Lee Felsenstein, one of the designers—if not the major designer—of the unit, was there answering the criticisms. Yes, he thought that the production models would have a larger screen. And, yes, they might be a little lighter in weight. I forget how many computers Lee has been involved with, but I remember seeing him a long, long time ago when he was putting the finishing touches on the Astral 2000 computer. It got a beautiful write-up in *Byte*, but I never saw one on the market. He then designed the SOL for Processor Tech. Maybe Lee will hit a winner one of these days—perhaps he has with the Osborne.

Few of the major firms in our business bothered to show at the Faire. For the most part there were smaller firms—and a lot of software. By my count, three of the top ten firms were there.

The attendees seemed to be almost entirely computer hobbyists, which is probably why the major firms didn't bother to spend the money. Some exhibitors said they were happy with the cash flow at the show, but most of the exhibitors were disappointed. The recession, at least in San Francisco, seems to have hit the micro field. By the time you add up the cost of air fares, hotels, car rentals, meals, booth space, booth exhibits and decorations, literature, shipping costs for the exhibits, people to man the exhibits and other expenses, it is unlikely that many exhibitors showed a profit. Mostly it seems to be an ego trip, which probably explains the preponderance of small firms.

Sherry and I collected literature from every booth we could. I've been looking this over and I can see why research shows that about 98 percent of the show literature is thrown out after a show and never read. Little of it is readable. I don't know what to do about that. There is an incredible need for some literature workshops to help manufacturers design and prepare their literature. Yet where can you run such a workshop? Exhibitors—the people who need to know this—are too busy during shows to attend a workshop. And where else are they together in one spot?

If anyone ever looks over literature from the viewpoint of the businessman who has no background in computers, he will find that virtually all show literature is wasted. Of course, much the same holds for ads in the magazines. I am planning a magazine for this readership for the fall, but where am I going to find advertisers whose literature the readers can understand?

While some of the booths at the Faire are getting more professional, a large number of them still feature card tables and signs written in Magic Marker. One of my fellow publishers, a holdout from the T-shirt and dungaree days of micros, runs one of the junkiest booths. If we ever do manage to set up a micro show which attracts businessmen in any numbers, we'd better learn to dress for success instead of the commune.

The PMC-80 seems to be progressing well, with their fast loader now a part of the system for really speedy cassette loading. They have been joined now by another TRS-80 software-compatible system, the LNW-80.

One surprise at the show was the absence of any Japanese entries. I suspect that they are still trying to figure out how to get into this market. This is a very big question for them. They know there is one hell of a market here, if they can get into it. Most of the people who know the answers are gainfully employed in the industry and not at all likely to whisper the secret, even if they know it. Perhaps we will see more at NCC.

Infoworld put on a cocktail party for the survivors of '76—the people who have managed to stay in the business for five years, which is a long time for this industry. I did spot most of the old-timers at the gathering. Some snacks were served, but there were so many people that the whole feeding part of the party was over in seconds. It seemed likely that more food would be brought, but it wasn't, so most people went hungry. John Craig got up to read a cute piece he'd written about the pioneers of the field, but he was unable to get the crowd quiet enough for many people to hear it.

The next evening I had dinner with Bill Godbout, George Morrow and Heath Kline (who runs those awful ads in another magazine). Both Godbout and Morrow make boards for people interested in the leading edge of micro technology—S-100 equipment. These chaps are so deeply into the industry that they know

META TECHNOLOGIES

161

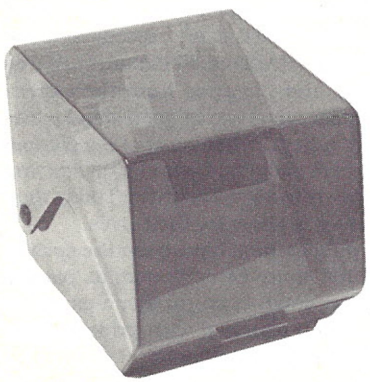
26111 Brush Avenue, Euclid Ohio 44132

CALL TOLL FREE 1-800-321-3552 TO ORDER

IN OHIO, call (216) 289-7500 (COLLECT)

FILE BOX

DISKETTE STORAGE SYSTEM



$24.95 for 5¼" disks

$29.95 . . . for 8" disks

MTC brings you the ULTIMATE diskette storage system, at an affordable price. Storing 50 to 60 diskettes, this durable, smoke-colored acrylic unit provides easy access through the use of index dividers and adjustable tabs. Unique lid design provides dust-free protection and doubles as a carrying handle.

'RINGS' & THINGS

Help prevent data loss and media damage due to improper diskette centering and rotation with the FLOPPY SAVER™ reinforcing hub ring kit. 7-mil mylar rings install in seconds. Kit is complete with centering tool, pressure ring, 25 adhesive backed hub rings and instructions.

HUB RING KIT for 5¼" disks. $10.95
HUB RING KIT for 8" disks. $12.95
REFILLS (50 Hub Rings) . . . . $ 5.95

Protect your expensive disk drives and your valuable diskettes with our diskette drive head cleaning kit. The kit, consisting of a pair of special "diskettes", cleaning solution and instructions, can be used for 52 cleanings. Removes contamination from recording surfaces in seconds without harming drives.

CLEANING KIT for 5¼" drives . . . $24.95

PLASTIC LIBRARY CASES

(not shown)

An economical form of storage for 10 to 15 diskettes, and is suitable for your bookshelf! Case opens into a vertical holder for easy access.

5¼-inch diskette case $3.50
8-inch diskette case $3.95

Single Sided, Soft-Sectored 5¼-inch, (for TRS-80™) Mini-floppy

DISKETTES

$21.95 box of 10

These are factory fresh, absolutely first quality (no seconds!) mini-floppies. They are complete with envelopes, labels and write-protect tabs in a shrink-wrapped box.

PLAIN JANE™

DISKETTES

The Beautiful Floppy with the Magnetic Personality™

Thousands of people have switched to this low-cost alternative. These quality diskettes are packaged in a plain white box . . . no fancy printing, fancy names or fancy labels, not even our own (labels cost money). Trust us.

PLAIN JANE™ Diskettes $21.95
10 boxes of 10 . . . . (each box)$21.50

PLAIN JANE™ *Gold*

Introducing MTC's premium generic diskette. Single-Sided, Soft-Sectored, DOUBLE-DENSITY, 5¼-inch diskettes with reinforcing HUB-RINGS. Individually 100% ERROR-FREE certified. Invest in GOLD!

PLAIN JANE™ *Gold* $25.95

VERBATIM'S PREMIUM DISKETTES AT AFFORDABLE PRICES

DATALIFE™

Seven data-shielding improvements mean greater durability and longer data life. These individually, 100% error-free certified diskettes feature thicker oxide coating, longer-lasting lubricant, improved liner, superior polishing and more! Meets or exceeds IBM, Shugart, ANSI, ECMA and ISO standards. Reinforcing HUB RINGS help prevent data loss and media damage, reducing errors.

Buy the best . . . buy DATALIFE™

VERBATIM DATALIFE™ DISKETTES

5¼-inch (box of 10)
MD525-01 . . . . $26.95
10 boxes of 10 . . . . (each box)$25.95

8-inch FLOPPIES
Double-Density, FD34-8000 . $43.95

MICROPARAPHERNALIA

DISKETTES (box of ten)

5¼" PLAIN JANE™ $21.95
5¼" PLAIN JANE™ *Gold* $25.95
5¼" DATALIFE™ MD 525-01 . . $26.95
8" DATALIFE™ FD34-8000 . . . . $43.95

NEWDOS by APPARAT

NEWDOS/80 by Apparat $149.95
NEWDOS + to NEWDOS/80 UPGRADE . . . . . CALL
NEWDOS + with ALL UTILITIES
35-track $69.95
40-track $79.95

BOOKS

TRS-80™ DISK AND OTHER MYSTERIES . . $19.95
MICROSOFT™ BASIC DECODED $29.95
1001 THINGS TO DO WITH YOUR PERSONAL COMPUTER . . . . $ 7.95

Let Your TRS-80™ Test Itself With

THE FLOPPY DOCTOR & MEMORY DIAGNOSTIC

by THE MICRO CLINIC

A complete checkup for your Model I. THE FLOPPY DOCTOR completely checks every sector of 35- or 40-track disk drives. Tests motor speed, head positioning, controller functions, status bits and provides complete error logging. THE MEMORY DIAGNOSTIC checks for proper write/read, refresh, executability and exclusivity of all address locations. Includes both diagnostics and complete instruction manual.

SYSTEM DIAGNOSTICS $19.95

An improved version of the SYSTEM DIAGNOSTICS above. Designed for single or double density, 35-, 40-, 77-, or 80-track disk drives. Includes new and modified tests. Features THE FLOPPY DOCTOR, Version 3.0.

SYSTEM DIAGNOSTICS-V3 . . $24.95

CALL FOR INFORMATION ON OTHER PRODUCTS

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation. DATALIFE is a trademark of VERBATIM. PLAIN JANE, AIDS-I, AIDS-III, CALCS-III, CALCS-IV, MERGE-III are trademarks of MTC.
©1981 by Metatechnologies Corporation, Inc.

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

Products damaged in transit will be exchanged.

PRICES IN EFFECT June 1, 1981 THRU June 30, 1981,

Prices, Specifications, and Offerings subject to change without notice.

8106

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6½% sales tax.

On my trip to the West Coast Computer Faire, I had dinner with (left to right) Heath Kline of Priority 1, Bill Godbout of Godbout Electronics and George Morrow of Morrow Designs. Note the small stereo recorder in the foreground. Copies of the entire dinner conversation can be had for $500, and are a bargain at that price.

everyone and all of the scandals. It isn't often that I have dinner with people who have enough interesting things to say so that I shut up—I didn't get a word in edgewise.

The Faire? You didn't miss much.

Career Growth

The microcomputer field is growing at about 300 percent per year. Have *you* put yourself in a position to grow as fast (or faster) than the industry? It's all too easy to let the world go by while you sit still.

In terms of a career in microcomputing, this means that you want to look for some medium to smaller firms, ones that seem to have the best chance at keeping up with the growth. If you get boxed into a McGraw-Hill or an ITT, your chances of getting into any position of real authority are minimal. Big companies operate on the decisions of higher-level managers, who are most reluctant to accept either change or risk. It can be an extremely frustrating experience for a person with ideas, or for the entrepreneurial chap who wants to make things happen.

A lot of firms in the micro field have splendid prospects, and most of them are looking for people who are willing to work hard, have ideas and show a willingness to learn. I'm sure looking for that sort of people. In talking with the managers of other growing firms, I find that my needs are the same as theirs. A bright person, willing to make things happen, is most welcome.

In my firm, I need people to help sell ads for the magazines, and to staff the new *Desktop Computing* magazine I'm planning for the fall with editors, writers, circulation people and so on. Circulation sales are complicated, calling for skills in writing, direct mail and advertising, and in dealing with reps, newsstand distributors, stores, shows and clubs. Everything you do calls for a knowledge of the field plus careful measurement of results.

We need DP people to help set up new and more intricate data handling and communications systems. We need people to help Instant Software in the areas of marketing, advertising, promotions, packaging, duplication, mastering, evaluating, special programming, conversions for other systems, dealing with hardware firms, packing and shipping, evaluating the products of other software publishers, copyright protection, legal management, financial management, accounting, building maintenance, selling at shows, laboratory maintenance and repairs of hardware, point of purchase displays, automated material handling, materials purchasing and testing, quality control . . . it seems endless.

If you are interested in growth (and you don't smoke), we're certainly growing. Instant Software alone is expected to grow about 500 percent during the next year. *Desktop Computing* is aiming at a circulation of over 100,000 within the first year. Both *80* and *Microcomputing* are aiming at about 100 percent growth during this year. The book department is also looking at 500 percent or more growth.

Peterborough is a small town (4500) in a little isolated pocket which has so far avoided most of the incredible growth of the eastern part of the state (DEC, Wang, etc.). Most people live in the tiny surrounding villages, in a rural atmosphere, with Peterborough the shopping and business center. Housing prices are still very reasonable by California standards.

The quality of life in New Hampshire is without equal in our country—perhaps in the world. Taxes are minimal, the state is over 80 percent wooded, and the air is crystal pure. We're within easy driving distance of Boston for travelling and cultural events.

If you're interested in a career instead of just a job, write and tell me what you think you can do for my growing conglomerate, citing past experience to back yourself up. Money still talks, so be candid about salary requirements.

But, whether you are interested in the beauties of New Hampshire or the rigors of Silicon Valley, don't let yourself get into a spot where you are not able to grow with this incredible industry. Whether you are into the hardware, the software or any other aspect of micros, keep learning and developing your abilities and your career.

When you feel that you are being held back, look around. Be careful not to get sucked in by a big jump in salary to some position where you will no longer be able to have the freedom to learn and the right to make decisions. The lure of large companies has brought down many a good man. Yes, I tried it and discovered the frustrations involved. Fortunately, it was early on and I learned from the experience. That stood me in good stead when leaving college, when most of my compatriots were seduced by fantastic starting salaries, from big companies. Today they have little to show for their lives.

My publications have been the springboard for many careers, and, as painful as it may have been (in some cases) to lose them, I watched with pride as they moved on to bigger things.

Articles

A number of opportunities present themselves to those who prefer to work at home, including programming, program conversions and writing articles or books.

There is almost no limit to the need for informative articles. *Microcomputing* wants articles that will help the owners of all types of microcomputers on using different types of printers, on evaluations of programs, on networking systems, on uses in business, in education, in the home, for science, machine control, telecommunications, simulations . . . it's endless. *Microcomputing* is for the relative newcomer, so take it easy on the buzzwords. If you can only write for computer scientists, perhaps *Byte* is your medium.

My new magazine will need articles written for the businessman, totally without technical jargon. Businessmen already know that computers are going to help them and save them money, but they haven't the time it takes to become experts, so they need a magazine which will tell them, in English, what computers can do these days. We want articles written by the actual users, by dealers who have managed successful installations, by programmers with satisfied customers—or even by manufacturers with installations of which they are proud. Businessmen want information—straight information.

It isn't that hard to translate bytes into an equivalent of ability to store a certain number of names and addresses, or pages of text. Printer speeds can be expressed in how long it takes to type a one-page letter rather than characters per second,

lines per minute, etc. It will be an interesting experience for many writers to see if they still have some ability to write in English.

Authors should remember that we live in a publish or perish civilization. This means that when you go to look for a new position, every article you've had published is money in the bank. Many authors tell us that each article is equivalent to about $1000 per year in added pay when they change jobs. Dealers will find that articles about them are a key to getting more customers. Consultants gain considerable credibility when they are published.

It is odd that so few firms in our field have managed to get articles written about their products. The consumers are desperate to know what really works—and they don't have to turn far to get an earful of horror stories.

One more thing. We pay right up front for all articles, upon acceptance.

Classes in Shambles

Recent studies reveal the sad plight of teachers in today's educational systems. Classes are in a shambles. It is becoming almost impossible to teach because an ever-increasing number of students have lost interest and are distracted by those youngsters who are there against their will. It is no wonder, then, that teachers are deserting the field by the tens of thousands.

When I see movies of teachers trying to cope with children who are uninterested in learning, I wonder why these teachers aren't pushing as hard as they can for computers to take over their thankless jobs. With individual computers for each student there would be less opportunity for students to create havoc, throw erasers (there will be no more erasers) or molest the teachers.

Through computerized teaching systems we will eventually be able to provide a better quality of education than is currently available in most schools. Courses can be made interesting and be taught by some of the best teachers in the world. The computer, as I've written before, may be used to adjust the speed of the teaching to the interest and intellect of the student, rather than just for drills of data to be committed to memory.

Once we have been able to develop better teaching systems, not dependent upon teachers for other than some minor support and advice, it may be possible to move these systems into the home—much as we expect work stations to be able to move into the home once we have adequate communications systems to support that mode. This may be a healthy change for society and, in the long run, result in better-motivated children.

There are many factors which have brought about the changes in schools over the last 40 years. We can certainly point fingers at the almost standard two-parent working family which is now needed just to pay for homes and food, but has reduced the influence of the mother in the home during after-school hours. Worse, the ubiquitous TV not only claims our children, but almost totally occupies the parents in the evening too, removing the chances for any serious communication between parents and children—other than sharp demands to temporarily solve problems.

Then we also have that steady diet of violence on television. There is no way that can be helpful. Add to that the doctrine of Dr. Spock and the "let the children do what they want" attitude, and we may have the roots of the lack of respect, the graffiti, the vandalism which is now engulfing us.

One of the problems of compulsory education is that it not only forces kids to be in school when they are uninterested—and are thus prone to interfere with those who are—but it also acts upon the teachers, who have captive audiences and thus no responsibility to make the work interesting.

If education is to succeed it is going to have to realize that it is show biz and accept that it is going to have to be interesting. It really *can* be done.

What about kids who are *not* interested in education? Isn't it enough that we make it available? Taking the horse to water, as it were, but agreeing that we don't have to make it drink?

We're prisoners of our past in much of this. Schools developed for some good reasons. They evolved into giant baby-sitting systems for most families, keeping kids out from underfoot—and out of the streets, at least for most of the day. Summer vacations may have been a blessing for the kids, but many mothers sighed when they came.

Perhaps we should legitimize child labor, get rid of the minimum wage requirements for children under 18 and see what happens. Oh my bleeding heart! Would kids find that the alternatives to school are less pleasant than school?

One of the problems, obviously, is the concentration of the poor which has built up in most of our cities. These people came into town from the farms a generation or two ago and have been dragging our whole society down. The answer to most of their problems is education and motivation, but no one knows how to get those started for this group. The problem is killing the cities as well as poisoning our country, so some solution must be found.

I don't know whether or not automated education will help those who need education the most. Can some means of motivation be found? Well, we *have* to do something. In the meanwhile, automated education should help make information and understanding of our world available to those who do have the motivation and who let themselves be helped by this resource. It will enable them not to be dragged down by those who are unmotivated and who are trying to prevent others from learning.

It is my intention with Instant Software to pursue the development of automated learning systems. I have in mind, at present, something akin to a video cassette which is coupled to a microcomputer. The micro would be programmed to show the video program and stop every now and then to make sure the student is keeping up with the material. If the student is unable to answer questions correctly, the machine would rewind and go over the material again, ending up with a different series of questions. In this way the progress of the course would be matched to the student.

If this should catch on, I can see a time when we might be using video disks. Even if a particular program were to cost several million dollars to produce, it would be a bargain because we could use it to teach hundreds of millions of people. It could be translated into any language and provide a relatively low-cost education, even for the most remote and poorest of the third-world nations.□

By Robert W. Baker

ROM Packages From Skyles

Command Repertoire Expanded

Here's some information on two interesting ROMs from Skyles Electric Works:

Command-O

Command-O is a 4K ROM for the 8016/8032 computers that adds 19 commands to BASIC. Eight of these commands are for program editing and debugging. The remaining commands provide powerful BASIC commands for screen formatting and disk file manipulation. Most of the commands can be used in direct or indirect statements. Here's a quick description of the commands provided:

INITIALIZE sends the initialization command to the disk to initialize one or both diskettes. The initialize command is not supported by BASIC 4.0 and would normally have to be sent via the command channel (channel 15). This command eliminates the need to use the command channel, and thus does not waste disk buffer space.

MERGE loads a program from disk and places it at the end of the current program in memory. If a specific line number is indicated in the MERGE command, the program is stored starting at that line number instead of at the end of the program. This lets you use program overlay techniques easily. Valid line numbering is the user's responsibility, since the merge command does not make any check on the line numbers in either program.

MOVE moves the cursor to a specific row and column position on the screen.

EXECUTE loads and then runs a program from disk.

SCROLL turns on the paging window, the enhanced screen editor and the softkey features (described later).

OUT disables the features enabled by the SCROLL command.

SET predefines a specific keystroke sequence of up to 190 keystrokes for any one specific key on the keyboard. This softkey feature is similar to Program Function (PF) keys on larger computer systems. It allows redefining any key as a common or heavily used command sequence. This can be a great time-saver and a real convenience once you get accustomed to using it.

KILL disables the ROM software and all related functions.

SEND sends a command directly to the disk without opening the command channel and tying up disk buffer space. This can actually be used to send an IEEE command to any IEEE device.

PRINT USING is not a full function PRINT USING but is still very powerful and extremely useful in the form provided. A string variable defines the print format using number signs (#) to define character positions (or fields) within an 80-character line. Numeric data can be formatted with separating commas and a fixed decimal point, with up to ten digits maximum. It does not support floating dollar signs, leading zeros and additional functions commonly provided by PRINT USING commands on other systems. Also, you cannot have fixed text within the format specification. Only #'s are allowed to define the fields that are to be substituted.

BEEP provides an easy method of generating sounds using the CB2 line of the user port. The command allows setting the duration and tone of the desired sound.

AUTO provides automatic line numbering when entering BASIC program lines. It allows specifying the starting line number and the step value to be used between lines.

DUMP displays a list of all nonarray variables. It will indicate all simple string, real and integer variables.

DELETE allows deleting a range of program lines that are expressed just like when using the LIST statement.

FIND locates and displays a BASIC program line containing a specified string or statement. Optionally, a specific line range can be examined instead of searching the entire program.

HELP requests help about an erroneous line. The line in error will be displayed with a reverse field marker where the error most likely occurred.

TRACE enables the tracing function for tracing and single stepping through BASIC program lines. When enabled, the current line being executed is displayed at the top of the screen.

OFF disables the tracing function.

RENUMBER provides a handy line-renumbering utility. You can specify the starting line number, the increment value and even a line range to be renumbered if desired.

The enhanced screen editor provided by Command-O lets you scroll up and down through a program listing by using the cursor-up and cursor-down keys. You don't have to continuously LIST segments of the program that will fit on the screen. Just list any part of the program, then scroll to another desired section if not currently on the screen. Another feature of the new editor is repeating keys. Any key held for more than a half-second will repeat at the rate of 15 characters per second until the key is released. This occurs for all keys on the keyboard, including all cursor movement keys.

The 8016/8032's new control characters are now generated at the touch of a key with Command-O activated. The Command-O editor transforms the "ESC" key into a control key. Any key pressed (A–Z) while the ESC key is held generates the appropriate ASCII control key value. Thus, if you hold ESC and press G, it generates a control-G code and

Address correspondence to Robert Baker, 15 Windsor Drive, Atco, NJ 08004.

IF YOU CAN WAIT A MINUTE, WE CAN SAVE YOU $1,000.

With the Starwriter™ Daisy Wheel 25 cps printer from C. Itoh.

A business letter, written on a 45 cps word-processing printer, might take about two minutes to print.

With the Starwriter, it might take closer to three.

The typical 45 cps printer retails for about $3,000.

But the Starwriter 25 retails for about $1,895—thus saving you about $1,000.

And therein lies the biggest difference between the Starwriter 25 and the more expensive, daisy wheel printers.

The Starwriter 25 comes complete and ready-to-use, requiring no changes in hardware or software. It uses industry-standard ribbon cartridges, and it's "plug-in" compatible to interface with a wide variety of systems, to help lower system-integration costs.

Using a 96-character wheel, it produces excellent letter-quality printing on three sharp copies with up to 163 columns, and offers the most precise character-placement available, for outstanding print performance.

C. Itoh's warranty;

3 months on parts and labor, supported by one of the best service organizations in the industry.

$1,000 OFF

Leading Edge Products, Inc.,
225 Turnpike Street,
Canton, Massachusetts 02021

Dear Leading Edge:
I'd like to know more about the Starwriter, and how spending a minute can save me a grand.
Please send me the name of my nearest dealer.

Name ____________________

Title ____________________

Company ____________________

Street ____________________

City/State ____________ Zip ______

Phone: Area Code ____________________

Number ____________________ KM-6

LEADING EDGE. 355

Leading Edge Products, Inc., 225 Turnpike Street, Canton, Massachusetts 02021
Dealers: For immediate delivery from the Leading Edge Inventory Bank™ call toll free 1-800-343-6833
In Massachusetts, call collect (617) 828-8150. Telex 951-624

sounds the bell. The new control characters supported are:

- scroll up or down
- insert or delete entire lines
- delete characters to the left/right of cursor
- select Text or Graphics mode
- ring the bell
- define the screen window

Another nice feature of the enhanced screen editor is the ability to "eat" characters to the right of the cursor. When the stop key is pressed the cursor stays in the same position and the character to the right of the cursor is deleted. The remainder of the characters on the line are drawn to the cursor to replace the deleted character.

The addition of the Command-O ROM does add many useful and powerful commands to the standard 8016/8032 system. Keep in mind, though, that whenever activated, there can be a significant increase in execution times when running BASIC programs. Also, programs written with the new commands will not be portable to other systems unless they also have the Command-O ROM installed. However, the debug and editing features are extremely handy. Also, remember that you can enable and disable the ROM at will, with complete control of when the features are active.

Disk-O-Pro

The Disk-O-Pro ROM is for older PET/CBM systems that haven't been upgraded to the BASIC 4.0 ROMs. Disk-O-Pro adds all the BASIC 4.0 disk commands plus some new ones, and gives you access to the Programmer's Toolkit as well (if you own one).

The ROM normally mounts in the empty UD3 socket that is addressed starting at $9000 hex. There's even special mounting hardware for all known PET models and various expansion hardware configurations. Even though Disk-O-Pro will function with all versions of the Commodore disk (2040-DOS 1.0; 4040-DOS 2.1; or 8050), the 2040 disk with DOS 1.0 cannot support all Disk-O-Pro commands.

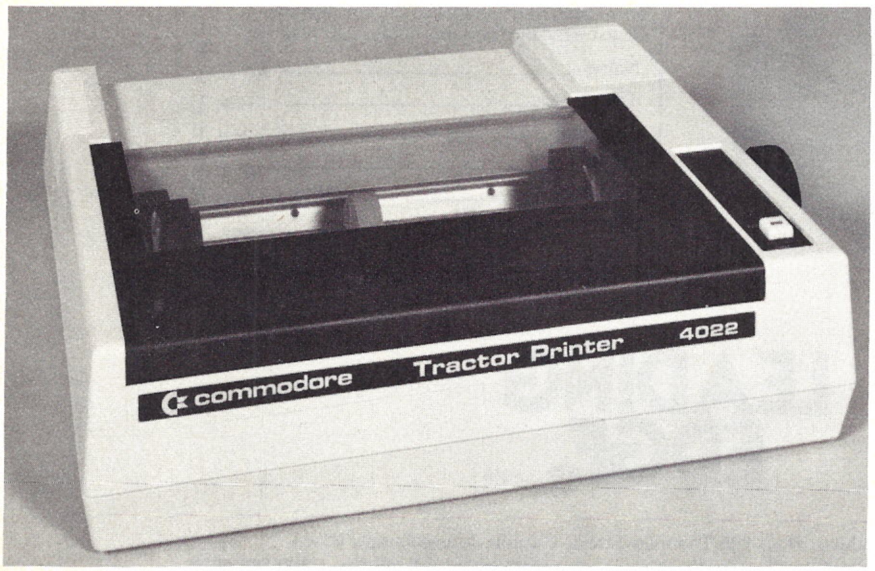
Table 1 shows the Disk-O-Pro commands that are in addition to those provided by the Programmer's Toolkit if available.

The BASIC 4.0 disk commands function as described in the Commodore documentation, use the same command syntax and generate the same token values within programs. The remaining commands are the same as the corresponding functions provided in the Command-O ROM and described previously.

The documentation provided with the Disk-O-Pro ROM is an excellent 84-page manual written by Gregory Yob. It provides plenty of examples, information on the internal workings of the commands, hints on using hidden features and cautions or warnings where appropriate.

All Disk-O-Pro commands can be executed as direct or indirect statements, but, as mentioned in the manual, several precautions must be observed. The ROM must be enabled to correctly handle any program using the added commands. Disk-O-Pro converts the new commands to special token values that are normally unused by BASIC 3.0. The ROM must be enabled to properly decode these token values for any function: editing, listing, executing, etc.

Previously, many of the ROM-based software additions were used primarily during program development, but were not needed after the program was completed. Disk-O-Pro, on the other hand, actually participates in running a program. This means the program will run slower and the same program may not be able to run in a system which does not have Disk-O-Pro installed.



The new 4022 tractor feed printer.

KILL	HEADER	RENAME
BEEP	CATALOG	SCRATCH
SCROLL	DIRECTORY	MERGE
OUT	BACKUP	CONCAT
SET	COPY	DOPEN
PRINT USING	COLLECT	DCLOSE
INITIALIZE	DSAVE	APPEND
SEND	DLOAD	RECORD
EXECUTE		

Table 1.

On the other hand, compatibility with the new BASIC 4.0 is quite good. Disk-O-Pro uses the same token values and follows many of the internal 4.0 procedures. There's a good chance that a program using these features would be portable between systems. You should be aware, though, that there are a few minor differences that could possibly cause problems. The special variables DS and DS$ are provided and make error checking on the disk a very simple task. BASIC 4.0 checks the disk status only when the status variables are referenced. However, Disk-O-Pro checks the disk status with every normal disk activity and stores the disk status as ordinary variables. Thus, the DS and DS$ values will be lost after a program change, a CLR or a LOAD.

Another area I should point out is that there are a few things provided by BASIC 4.0 that Disk-O-Pro can't handle. The changes to BASIC are too fundamental for an add-on package to emulate. The garbage collection delay is still there with Disk-O-Pro. Also, the problem with line feeds after a carriage return in a PRINT# statement still exists. With Disk-O-Pro you'll still have to use the familiar PRINT# . . . . ;CHR$(13); statement when writing to disk. There are a few other minor differences concerning the IEEE bus, but these shouldn't matter to most users.

Remember that Disk-O-Pro does offer several features above those provided by BASIC 4.0 but that these cannot be used by another system unless the Disk-O-Pro ROM is installed. Also, with the Disk-O-Pro ROM you don't lose the capability of using the Programmer's Toolkit. If you upgrade to BASIC 4.0 you'll not be able to use the Toolkit with the new system ROMs. All in all, the package is very nice and has its good and bad points. Its value to you depends on your needs and system configuration. Also, remember that you do have control over whether or not the ROM software features are active.

For more information on either ROM package, Command-O or Disk-O-Pro, you can write: Skyles Electric Works, 231E South Whisman Road, Mountain View, CA 94041.

Educational Software

Teaching Tools: Microcomputer Ser-

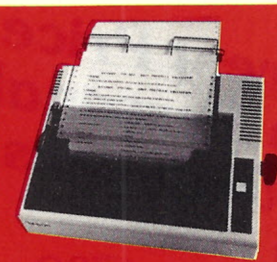
like the golf pros

THE STOCKING SOURCE

comes through with lower prices

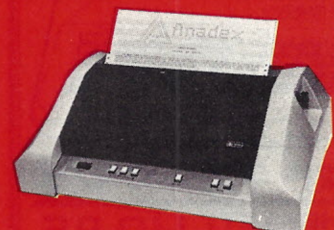
- ***Your Closest Link to the Manufacturer***
- ***All Items Listed are Stocked***
- ***We Support & Service Products Purchased From Us***

NEW EPSON Model MX-80FT List $745.



Friction Feed has been added to the popular MX-80. All other features the same, including an adjustable tractor.

Model MX-80 . . . . List $645.



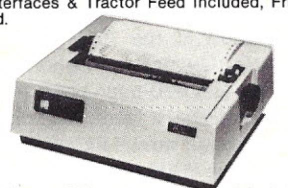
Anadex Model DP-9001 . . . List $1550.
80 Column

Model DP-9501 . . . . List $1650.
132 Column

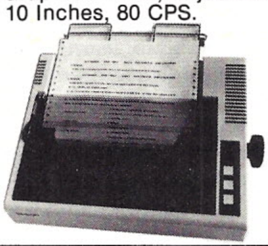
Unbelievable Dot Matrix Printer with All Functions Program-Accessible. Has 4-Character Font, Hi-Res Graphics, Switch Selectable Parallel, Serial & Current Loop Interface, Operator Selectable Protocal.

PRINTERS

Okidata —
Microline 83 . . . . List $1260.
Up to 15" Paper Width, 136 Columns Standard, 132 Compressed, Vertical Tab, Top of Form, Parallel & RS-232C Interfaces & Tractor Feed Included, Friction & Pin Feed.



Microline 82 . . . . List $799.
Microline 80 . . . . List $599.
Tractor Feed for Microline 80 & 82 . . . . . . List $50.
Hi-Speed Interface for Microline 82 & 83 . . List $210.
256 Character Buffer, Expandable to 2K
RS-232C Interface for Microline 80 . . . . . . List $150.
256 Character Buffer
Epson — Model MX-70 . . . . List $495.
Bit Plot Graphic Printer, Adjustable Tractor 4 — 10 Inches, 80 CPS.



NEW—INTERFACES

Hi-Res Bit Plot Graphics for Epson Printers . . . . List $90/Set.
Plug In Prom Set
Vardon 1K Serial Interface Buffer for Epson MX-80 . . . List $175.
(Complete with Cable)
Tymac — Apple Interface Using 8th Data Bit . . . . List $139.
(Complete with Cable)

INTERFACES

Apple Plug-In Interface & Cable . . . . List $110.
Serial Interface & Cable List $90.
IEEE 488 Interface & Cable . . . . List $80.
TRS-80* Expansion Interface Cable . . . . List $35.
*A trade name of Radio Shack — Tandy Corp.
Model 232 Universal Serial Interface Cable . . . . List $25.

master charge VISA

No Penalty for Credit Card Orders

NEW—PRINTER LINES

Centronics Printers In stock.
Model 730-3 (Serial) - List $845.
Model 737-1 (Parallel) - List $995.
Model 737-3 (Serial) - List $1045.
Model 704-9 (Serial) - List $1795.
Model 704-11 - List $1870.
Also Ribbons & Accessories
All other Centronics Printers are Available.
We now stock NEC Spinwriters, Models 5510, 5515, 5520, 5525 & 5530. Also Thimbles & Ribbons. All other models are available, including Band Printers.

TEST EQUIPMENT

We stock Beckman, Fluke, Global Specialties, Keithley & NLS DMM's and probles. Also Hayes Technical Breakout Boxes.

COUPLERS & MODEMS

— Tek-Com

FURNITURE & PRINTER STANDS

Systems Furniture and Printer manufacturers

ASK FOR OUR INSTANT DISCOUNT
From Roy Hawthorne
Talk To Bill Tokar On Applications

CALL TOLL FREE
U.S.A.
1-800-521-2764
MICHIGAN
1-800-482-8393

Reminder:
We are open 8:30 to 6:00 PM EST Monday through Saturday

WRITE TO:
"The Stocking Source" ✓288
24069 Research Drive
Farmington Hills, MI 48024
313-474-6708

vices recently announced a small but interesting collection of educational programs for the PET. The programs were developed by educational psychologists and extensively tested in classrooms.

A Match Game is useful for memory building, and for putting lesson reviews into a game format. The game can be played with either exact matches or paired matches. Users can choose from the many options built into the program (shapes, math and synonyms) or enter their own items. The size of the game board can be varied and one to four players can participate.

The Letters & Numbers program is ideal for young children first learning to use the computer. It uses upper- and lowercase letters and numbers created with the PET graphics. Options within this program include matching of one or more items, completing sequences and filling in missing items. In each option, letters, numbers or both can be selected.

The Addition and Subtraction programs are suitable for grades one through six and for special education classes. Problems are presented vertically and answers are input from right to left just as you would think through the problem. For added convenience, carrying or regrouping (borrowing) can be marked in the problem. Immediate feedback is given, with graphics reinforcing correct responses. By making selections from a menu, the user can select the number and level of difficulty of the problems, set a time limit or the number of attempts allowed on each problem, and control several other aspects of the program. The addition program has 24 levels of difficulty while the subtraction program has 12 levels.

The sample copies of the Addition and Subtraction programs received for review were excellent. In fact, the Addition program was awarded second prize in the Texas Instruments Author Incentive Program. The programs were completely user-proof. Pressing inappropriate keys does not affect the program, except for displaying a "STOP IT" message. Also, the PET stop key is disabled to protect the program.

The problem displays are wide-spaced for easy reading. The number zero (Ø) is replaced with the letter "O," since it was found that some children found zero (Ø) and eight (8) hard to distinguish on the PET. The problems themselves are done in the way children are taught to do them; the child does not have to adapt to the computer. In addition, the child receives immediate feedback. Correct answers are reinforced by cute robot figures using graphics, and the child's name is included in messages part of the time.

The teacher can preset the number of tries the child is given for each problem. When multiple tries are allowed, an incorrect answer is moved to the bottom of the screen with a "TRY AGAIN" message displayed. The most recent incorrect answer remains on the screen, lined up with the problem so the old and new answers can be easily compared. Special messages are used if the child is slow in responding or makes too many inappropriate responses.

The programs are especially suited for classroom use with special controls built in for teachers. For example, a teacher can set the level of the problems and a time limit. Each child can then work on the computer for the set amount of time without the teacher. The program cannot be stopped except by pressing Shift-@ at certain times. After each set of problems, various statistics are displayed to indicate:

- number of problems right on the first try and the average time per problem.
- number of problems right after the first try.
- number of problems wrong.
- total time for all problems.

I must say these are excellent pieces of software. Each program is supplied on cassette tape for $20 and versions are available for all model PET/CBMs. For more information, you can write Teaching Tools: Microcomputer Services, PO Box 12679, Research Triangle Park, NC 27709. In Canada you can contact SES (Computing) Inc., 267 Bain Ave., Toronto, Ontario M4K 1G2.

Book Review

I just finished reading a copy of the second edition of the *PET/CBM Personal Computer Guide* from Osborne/McGraw Hill. This is a 500-page book and lists for $15 in paperback form. The new edition now covers all model PET/CBMs, including the new 8000 series, BASIC 4.0 and DOS 2.0. It discusses the various model computers, the cassette drive, the 2040 and 8050 disks, and the 2022/2023 printers. The new edition covers just about every possible area of the Commodore system at the time the book was published.

This edition is a major revision of the original book published under the same title. The discussion of BASIC programming has been greatly expanded, and there are many sample programs and examples throughout the book. It covers everything from how to run programs already written to learning BASIC programming and programming techniques required for peripheral units.

The book is roughly divided into eight chapters, along with a number of appendices. Chapter 1 gives a basic introduction to the CBM computer. It shows what the CBM looks like, points out the differences between models and shows the various Commodore peripherals. The second chapter discusses how to operate the computer while the third chapter explains the screen editing features. The next two chapters discuss BASIC programming in simple terms and how to

make the most of the CBM features.

Chapter 6 covers the peripheral devices in great detail, including the tape, disk and printers. It covers all the features of each device with many examples on how to use each. The seventh chapter provides miscellaneous information for the more advanced user. It discusses how and where variables are stored, memory maps, assembly language programming, random access disk files, etc. The last chapter gives a detailed description of every BASIC statement and function, including the new 8000 series special functions.

Hidden in the appendices is a wealth of information, including detailed character code charts, notes on variations for the older Version 2 BASIC and detailed memory maps (both RAM & ROM).

Of special interest is Table F-3, which shows a list of all source code labels within the CBM BASIC ROMs for both BASIC 3.0 and BASIC 4.0. This one table can give you some indication where the various major routines are located in the BASIC ROMs, along with a cross-reference of the corresponding locations in both BASIC versions.

This book is definitely of interest and great value, regardless of your level of expertise. I should warn, however, that the latest edition has a large number of typographical errors. A second printing of this edition is supposed to fix these problems. There are also a few areas that are totally wrong and do not apply to the PET/CBM systems. The biggest problem areas concern the programming of disk files. Since the book was updated for BASIC 4.0 and DOS 2.0, the areas discussing BASIC 3.0 have a number of errors and should be used with caution.

Also, the book discusses a few items like program overlays and program line-formatting that do not apply to the PET at all. This type of logical error is planned to be fixed in a third edition scheduled for later this year. The next edition may also include more information, such as details on the machine-language monitor.

Even so, the current edition is still a great value and contains a good deal of information.

Miscellaneous

The publishers of *Compute* have announced a new magazine especially for the Commodore VIC computer. Under the name *Home and Educational Computing!*, the quarterly magazine was scheduled to start with the April-June issue, so it should be available by now. The material is intended to teach, entice and interact with readers to help everyone develop maximum benefit from the new VIC personal computer series. A special introductory price of $5 is available for the first three issues. For more information, write: Home and Educational Computing, PO Box 5406, Greensboro, NC 27403.

I finally got my 8050 disk from Commodore, so now I can review any programs that require the new high-density disk. If you have an older 2040 disk and have upgraded to DOS 2.1, you might want to try and get a copy of the new disk manual from Commodore. They now have a single user's manual for all 5-1/4-inch disks. The new manual covers the 2040, 3040, 4040 and 8050 disks and can be ordered as part number 320899. Unfortunately, I haven't been able to find the price.

The new disk manual includes descriptions of the programs supplied on the demo disks and the new disk commands in BASIC 4.0, along with handy quick-reference charts. Of special interest are the detailed descriptions of the disk header, BAM and directory formats for each disk type. This is definitely a very handy and valuable reference for any disk user.

For those who haven't seen the new nomenclature, 2040 is the original disk with DOS 1.0, 3040 is the European version and 4040 is the 2040 disk with the newer DOS 2.1 ROMs. By the way, now that I have an 8050 disk, I've created a new version of my DISK MASTER disk cataloging program for the 8050 drive. Contact me if you need more information.

LemData Products has announced a newer version of its PIE printer interface. Their original interface module was only designed to interface printers that used the standard Centronics interface. The PIE-C will interface just about any eight-bit parallel interface device to the PET IEEE bus interface. The new interface module mounts directly to the CBM with a custom enclosure.

The PIE-C provides complete address selection for device numbers 4 through 30. It works with WordPro and other programs that output true ASCII. There is also a CBM-to-ASCII code converter built in so you can print from BASIC or output program listings. All 256 codes are provided in either mode so you can use the printer control codes. Cost of the new module is $119.95 plus $5 shipping. For more information, write LemData Products, PO Box 1080, Columbia, MD 21044.

In case you haven't heard, Commodore has a new 4022 tractor feed printer available. Based on the popular Epson MX80 dot matrix printer, the 4022 offers the full CBM graphics set, reverse characters and many programmable features. Details and pricing were still sketchy at this writing, but this should be a very nice addition to the Commodore line.

Several people have reported various problems with the regular Epson MX80 printer's handling of upper/lowercase with the CBM. So far I haven't been able to find any more information on this but I'm still working on it. If you have any information that may be of help to others, please write.□

By Frank J. Derfler, Jr.

Revolution On the Wane?

Operator Enthusiasm Fizzling

The data communications revolution is rolling on, but have we reached a peak in the number of available bulletin board message systems? This month, I will discuss the rise and probable fall of microcomputer-based electronic message systems. I'll also look at what I think is the top terminal program for the TRS-80: Omniterm.

The Peak?

I recently contributed some background material for an article on electronic message systems (CBBS, ABBS, Forum 80, etc.) in the *Wall Street Journal*. I was happy to see the interest shown by this international publication in the commercial applications of what have been hobby systems. In the course of describing message systems to the *Journal* staff, it occurred to me that this phenomenon might not go on forever. This might seem like a strange thing to say when new systems are still appearing regularly, but the rate of growth has certainly decreased.

Please don't misunderstand—there has been no decline in the number of users of message systems or in the number of people interested in data communications. Modem makers and software sellers are still awash in orders, but the microcomputer-based message system rage is maturing.

It takes a considerable investment of time and money to run a message system. The average system operator is tied to his machine about half an hour a day simply viewing the screen for appropriate messages, backing up disks and compacting files. The disk drives get heavy use and maintenance can become a real concern. Several thousand dollars' worth of equipment may be dedicated to message system operation with no offsetting income or depreciation.

The thrill of being at the center of activity or the fulfillment of providing a service can wear thin quickly. Many systems come "on the air" with great fanfare, only to disappear with a gurgle and a sigh a few months later. The first wave of exuberant enthusiasts has broken on the beach and subsided.

The surviving system operators will be those with more practical or deeply-rooted reasons for running a system. They will also probably have some support in their work. Special-interest systems will continue to grow in some areas—although several of these with seemingly good potential have died already.

Systems sponsored by clubs in large metropolitan areas will do well. Systems sponsored by manufacturers (North Star, Microperipheral Corp., PMMI, etc.) should continue to be strong because of the service and communications with customers they provide. But otherwise, I predict the gradual demise of the simple message system run by an individual.

The increase of special-interest message systems on the information utilities (CompuServe and The Source) has created a form of competition for the privately-owned bulletin boards. The use of a tiny fraction of the information utility's processing power for a multi-user message system is more practical than the dedication of a microcomputer in many applications. The utilities can provide program transfer, large storage capabilities and other features found only on the most sophisticated bulletin board systems. The cost of the information utility is probably not as much of a limiting factor to growth as the lack of local circuits in many parts of the country.

Interestingly, there seems to be growing commercial use of microcomputer-based message systems for certain dedicated uses. Corporations can make use of sophisticated bulletin board systems to support research and development teams. Traveling marketing people can check into corporate systems at any time of day or night to pick up the next day's calls, deliver orders and run programs to figure commissions or discounts. Software maintenance and development personnel can exchange notes or even lines of code.

These people can appreciate and afford microcomputer-based systems because they avoid the problems of writing special software for large mainframe computers and the hassle of large system software documentation.

I can predict some parallel but separate trends. Strong centralized and well-supported systems will emerge. Software sophistication will continue to increase. Privacy and security needs will grow too. Corporate users may use more small systems, and private users may use more big ones. The microcomputer-based data communications era will emerge from the shake-out stronger and more useful than ever.

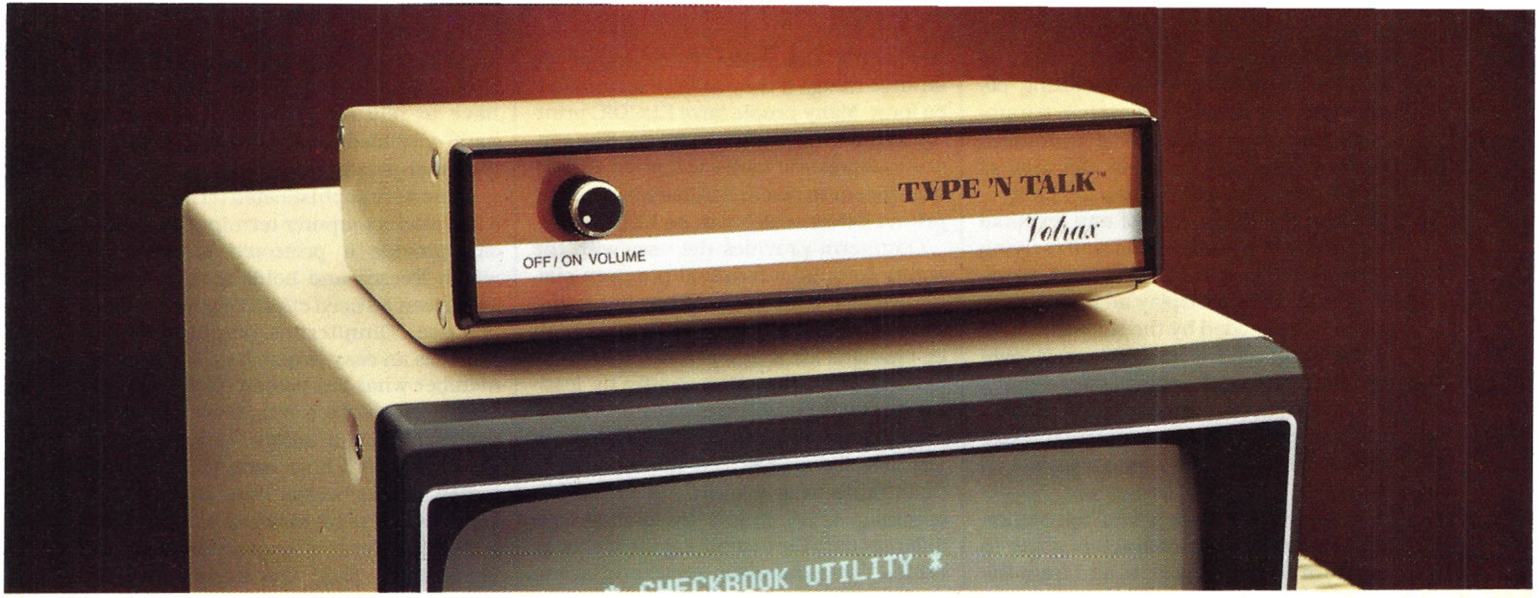
Omniterm

Last month, I mentioned a fine TRS-80 data communications program called Omniterm, written by David Lindbergh. It is available from Lindbergh Systems, 49 Beechmont St., Worcester, MA 01609.

Omniterm lets you use your TRS-80 as a smart data-communications terminal. It is a powerful program with many features. It takes a little experience to get the full benefit of all of the program's sophisticated options, but even a novice user can get on-line using the software in a matter of minutes. Let's look at the features that make the program so versatile.

Like other smart terminal programs we have reviewed, Omniterm will save the data received from an electronic message system or information utility on disk files. It will also allow the preparation of messages off-line for later retrieval from disk and efficient transmission out the serial port. It will provide a buffered output which sends all received data to a printer connected to the parallel port. The program will allow transmission of control codes and special characters such as brackets.

In common with the best terminal programs, Omniterm provides for the one-key transmission of a prestored block of data called a macro. This macro string is useful for the transmission of log-on codes, passwords and other frequently-used lines of letters and numbers. The macro characters are saved on disk and



TYPE-'N-TALK™ IS T.N.T.

The exciting text-to-speech synthesizer that has every computer talking.

- **Unlimited vocabulary**
- **Built-in text-to-speech algorithm**
- **70 to 100 bits-per-second speech synthesizer**

Type-'N-Talk,™ an important technological advance from Votrax, enables your computer to talk to you simply and clearly — with an unlimited vocabulary. You can enjoy the many features of Type-'N-Talk,™ the new text-to-speech synthesizer, for just $345.00.

You operate Type-'N-Talk™ by simply typing English text and a talk command. Your typewritten words are automatically translated into electronic speech by the system's microprocessor-based text-to-speech algorithm.

The endless uses of speech synthesis.

Type-'N-Talk™ adds a whole new world of speaking roles to your computer. You can program verbal reminders to prompt you through a complex routine and make your computer announce events. In teaching, the computer with Type-'N-Talk™ can actually tell students when they're right or wrong — even praise a correct answer. And of course, Type-'N-Talk™ is great fun for computer games. Your games come to life with spoken threats of danger, reminders, and praise. Now all computers can speak. Make yours one of the first.

Text-to-speech is easy.

English text is automatically translated into electronically synthesized speech with Type-'N-Talk.™ ASCII code from your computer's keyboard is fed to Type-'N-Talk™ through an RS 232C interface to generate synthesized speech.

Just enter English text and hear the verbal response (electronic speech) through your audio loud speaker. For example: simply type the ASCII characters representing "h-e-l-l-o" to generate the spoken word "hello."

TYPE-'N-TALK™ has its own memory.

Type-'N-Talk™ has its own built-in microprocessor and a 750 character buffer to hold the words you've typed. Even the smallest computer can execute programs and speak simultaneously. Type-'N-Talk™ doesn't have to use your host computer's memory, or tie it up with time-consuming text translation.

Data switching capability allows for ONLINE usage.

Place Type-'N-Talk™ between a computer or modem and a terminal. Type-'N-Talk™ can speak all data sent to the terminal while online with a computer. Information randomly accessed from a data base can be verbalized. Using the Type-'N-Talk™ data switching capability, the unit can be "de-selected" while data is sent to the terminal and vice-versa — permitting speech and visual data to be independently sent on a single data channel.

Selectable features make interfacing versatile.

Type-'N-Talk™ can be interfaced in several ways using special control characters. Connect it directly to a computer's serial interface. Then a terminal, line printer, or additional Type-'N-Talk™ units can be connected to the first Type-'N-Talk,™ eliminating the need for additional RS-232C ports on your computer.

Using unit assignment codes, multiple Type-'N-Talk™ units can be daisy-chained. Unit addressing codes allow independent control of Type-'N-Talk™ units and your printer.

Look what you get for $345.00. TYPE-'N-TALK™ comes with:

- Text-to-speech algorithm
- A one-watt audio amplifier
- SC-01 speech synthesizer chip (data rate: 70 to 100 bits per second)
- 750 character buffer
- Data switching capability
- Selectable data modes for versatile interfacing
- Baud rate (75-9600)
- Data echo of ASCII characters
- Phoneme access modes
- RS 232C interface
- Complete programming and installation instructions

The Votrax Type-'N-Talk™ is one of the easiest-to-program speech synthesizers on the market. It uses the least amount of memory and it gives you the most flexible vocabulary available anywhere.

Order now. Toll free.

Call the toll-free number below to order or request additional information. MasterCard or Visa accepted. Charge to your credit card or send a check for $345.00 plus $4.00 delivery. Add 4% sales tax in Michigan.

1-800-521-1350.

Votrax® 200

Distributed by Vodex
A Votrax Company — Dept. RT
500 Stephenson Highway, Troy, MI 48084
(313) 588-0341

Type-'N-Talk™ is covered by a limited warranty. Write Votrax for a free copy.

loaded in with the program at start-up along with other system operating parameters. These tables of system parameters lead to one aspect of Omniterm which makes it unique in the area of TRS-80 terminal programs—flexibility.

One of the major jobs of any terminal program is translation. Characters flying between the keyboard, serial port, parallel port and video display all must be encoded and decoded by the software. Most terminal programs provide their own unique keyboard decoding tables—that's how they can provide special characters and control codes—but they usually have only one or two tables which translate all codes.

Omniterm provides seven translation tables which handle characters going to and from the serial port, to and from the disk operating system, to the printer, to the screen and from the keyboard. These tables can change all characters, not just special characters and control codes. This means that Omniterm can speak in different data alphabets such as EBCDIC (but not Murray/Baudot; it can't handle the shift code). It also means that it has a nearly infinite ability to translate special control codes that might mean one thing to a video screen and quite another to a printer.

Some control codes meant for screen formatting can make your printer do strange things if they hit it in untranslated form. Many people have EBCDIC printers which they would like to use. The separate translation tables will allow the entire system to run in ASCII except for the printer driver which can be EBCDIC.

Omniterm provides the user with the ability to build a terminal program customized for both the local hardware and the remote system. This customizing can be changed for a new remote system or hardware configuration simply by loading in a different settings file from disk. A microcomputer can be changed from a terminal for an EBCDIC-speaking large mainframe to a terminal for an ASCII-speaking ABBS simply by loading in a different data file containing the alphabet tables, transmission characteristics, screen format parameters and macro listing. This system of translation tables and specification files gives great flexibility and customizing potential.

Another special feature of Omniterm provides great flexibility in the use of the buffer storage area. Buffer storage is simply RAM space that holds received data or data loaded in from disk files. In other programs, the buffer can be read only off-line or after the data in the buffer is stored on disk. Omniterm allows the user to look back in the buffer—even on-line—to read again what has already scrolled off the screen. This is a significant convenience for users of electronic mail and bulletin board systems. It gives an instant replay capability just as valuable as the one on televised football games.

One very real disadvantage of the electronic mail systems available to most users is the lack of any original note to work from when writing a reply. You usually have to rely on your own memory or hand-scribbled lines to recall the things the other person said.

Some very sophisticated terminals and a few microcomputer terminal programs can provide a protected screen area which freezes and holds received data. These can be used effectively with a little practice. Omniterm's scrolling buffer provides an easy way to help the user remember what has passed off the screen.

The screen of a terminal system using Omniterm has a unique look. The program "knows" how wide the terminal screen is and breaks lines that are too long at some point between words. Many programs have this wrap-around feature. These other programs usually deposit the leftover word(s) on separate fragmented lines. They then recognize the carriage return at the end of the fragmented line and begin a new line. This gives the screen a very choppy appearance and makes narrative material hard to read.

Omniterm will change carriage returns to a space if a line is too short and add its own carriage return when a line is too long. This feature eliminates fragmented lines and the short lines sent by many systems. An Omniterm screen is clean, full and well-balanced.

The Omniterm manual is also full and well-balanced. It has over 60 pages and even includes a glossary and complete index. A good manual is a must with a program this sophisticated, and David Lindbergh has done well.

Omniterm has many features which I

The Radio Shack Modem I is a direct connection modem providing originate and answer operation. It can be used as a standard RS-232-C connected modem, or with a special cable and software, it can operate with the TRS-80 Model I keyboard in the half-duplex mode. It sells for about $150.

don't have room to describe, but they include a large graphic picture of a bell that appears on the screen when a control G is received, a timer to tell you how long you have been on-line and complete command menus at every program level.

The program will run on a TRS-80 Model I with 32K of RAM and one disk. A version for the Model III should be available by the time this is published. Omniterm has my vote as the top TRS-80 terminal program available today.

Mailbag

Three quick items out of the mailbag this month: You have probably seen the new Radio Shack modem for the TRS-80. It was first described in a February flier. (Cat No. 261172, $149.95.) What they didn't say in the flier is that the device can be operated directly from a Model I keyboard without an expansion interface—but only in the half-duplex mode. You can address most bulletin boards or information utilities in half-duplex if you instruct them properly. (The D command on a CBBS or ABBS toggles between full- and half-duplex.) Half-duplex will not provide the constant check of operating conditions gained from full-duplex operation, but the price is hard to beat.

AMRAD, the Amateur Radio Research and Development Corporation, has published a good bulletin board list. Terry Fox and others in the club have put a lot of work into making the list as accurate as possible on the day it was printed. Drop $1 to AMRAD, 1524 Springvale Ave., McLean, VA 22101.

There is a big push on in California to force the telephone companies to provide TTY devices for deaf communications. These devices use Murray/Baudot code and cannot normally communicate with ASCII systems. Novation, Inc., has announced the development of a $650 portable combination terminal/telephone called infone. It has a store-and-forward-message capability, can use either Murray or ASCII and can actually "talk" in synthesized speech with a 250-word vocabulary for the visually handicapped. I will get more information on this development for next month's column, but look out—the future just tapped you on the shoulder.

Keep It Coming

If you operate a message system or market a data communications product, I would like to hear from you. Questions and comments are always welcome, but please include a stamped envelope if you expect an answer to paper mail. Send electronic mail to TCB967 on The Source, 70003,455 on CompuServe, or via the AMRAD CBBS 703-734-1387.□

Address correspondence to Frank J. Derfler, Jr., PO Box 691, Herndon, VA 22070.

✓See List of Advertisers on page 210

By Walter Koetke

Multiple Precision With Micros

Don't Let Large Numbers Throw You

More Multiple Precision Arithmetic

Last month we discussed multiple precision division and began to address the question of multiple precision multiplication. Listing 1 contains the program we developed to multiply an N-digit integer by a single-digit integer. To use this program, enter the numbers to be multiplied as data in lines 10 through 30. Line 10 contains the number of digits contained in line 20. Line 30 contains the single-digit number that multiplies the N-digit number of line 20.

The ability to adapt the algorithms of one program as part of another program is a valuable skill that all should develop. Last month's illustration of 3^{1690} was produced by a program written by a twelve-year-old who had discussed the program of Listing 1 as part of his class work in school. He extended the idea only slightly so he could produce tables of an integer raised to sequential powers. The program in Listing 2 will print the expanded form of $3^1, 3^2, 3^3, \ldots$ and on and on until the expanded form exceeds the number of digits indicated in the DIM statement of line 20. The limit of 3^{1690} imposed in line 40 is arbitrary.

Notice that the working portion of the program in Listing 1 (lines 100–160, 180) is identical to the working portion of the program in Listing 2 (lines 50–120). This student very nicely applied an algorithm learned in one situation to help him in another similar situation. He wisely made use of integer variables for all his computations, since this is a near-perfect example of a program in which the speed of integer arithmetic is an important asset.

The variables used are: N—the number of digits in the product; D(I)—the i^{th} digit of the product, counting from the right; M—the base; and K—the value of the exponent. All other variables are identical to those used in the program of Listing 1. Notice that the program can compute the powers of any single-digit number by changing the value of M in line 30 and the digit printed in line 150.

Even if students are too young to understand the concepts of this program, they are quite likely to be interested in the program's output. Obtain access to a printer, then calculate and print 2^1 through 2^{1000}, then 5^1 through 5^{1000}, then give the output to students and enjoy the results. You'll find students pursuing many different tangents from this inspiration, most of which will be clearly productive.

One rather predictable outcome of these programs is the desire to obtain the multiple precision product of an N-digit integer and an M-digit integer. This nontrivial problem is an excellent exercise for better students. For the majority of students, the importance of the mathematical concepts more than justifies two or three periods of class development of this program.

On the simplest level, we are going to develop a program to multiply A (containing N digits) times B (containing M digits) to obtain the product D (containing up to N + M digits). For example, we can multiply 31647 times 423 to obtain the product 13386681.

A traditional way of approaching the development of this program is to consider a specific example.

```
  31647—A
*   423—B
  -----
  94941
 63294
126588
--------
13386681—D
```

We would next try to generalize the procedures of the specific example. If we are successful, our generalization will work for any integer product. This approach will certainly work, but replicating the conventional hand procedures for multiplication requires considerable intermediate storage space.

In the previous example, A, B and D required 16 digits, but the three intermediate products also consumed 16 digits. For larger products, the use of intermediate storage far exceeds the storage used for the values in which we're interested. We are, therefore, going to look at multiplication in a somewhat different manner. Our new perspective will not require the use of memory for intermediate storage. As a result, your 16K microcomputer can calculate and print the *exact* product of two integers, each of which contains almost 2000 digits.

We can begin with the easy part. The numbers to be multiplied will be entered using the same method illustrated in Listing 1.

```
10 DATA 5                    number of digits in A
20 DATA 3                    number of digits in B
30 DATA 3, 1, 6, 4, 7        the actual digits of A
40 DATA 4, 2, 3              the actual digits of B
50 READ N,M
60 DIM A(N), B(M), D(N + M)
```

Each subscripted value of A will contain one digit of the number A. Similarly, subscripted values of B each contain one digit of the number B. The subscripted values of D contain not only digits of the final product, but also all digits of nearly all intermediate computation. Notice

```
10 DATA 3
20 DATA 5,1,7
30 DATA 4
40 READ N
50 DIM D(N+1)
60 FOR I=N TO 1 STEP -1
70 READ D(I)
80 NEXT I
90 READ M
100 C=0
110 FOR I=1 TO N
120 P=D(I)*M+C
130 C=INT(P/10)
140 D(I)=P-C*10
150 NEXT I
160 D(N+1)=C
170 PRINT "THE PRODUCT IS:"
180 IF C>0 THEN N=N+1
190 FOR I=N TO 1 STEP -1
200 PRINT D(I);
210 NEXT I
```

Listing 1.

MICROPROCESSOR SUPPORT I.C.'S

WE GUARANTEE FACTORY PRIME PARTS

We are going to become the largest supplier of prime microprocessor support I.C.'S. We guarantee that our I.C.'S are purchased from manufacturer authorized distributors. This is the only way to deliver prime parts at the lowest possible prices. Our committment is to offer the best price and the fastest delivery to our customer. We give many thanks to our valued customers who have helped us grow.

NEC 16Kx1 DYNAMIC RAM 200 N.S.

These are prime 4116's from one of the best MOS RAM manufacturers in the world.

4116	200ns
8 for $25.00	32 for 96.00

8080A	CPU	4.95	2708	EPROM	1Kx8	4.50
8085A	CPU	8.95	2716	EPROM	2Kx8	7.00
8086	CPU	99.95	2732	EPROM	4Kx8	19.00
8088	CPU	44.95	4118	STATIC	1Kx8	15.00
Z-80	CPU	6.70	4164	200ns	64Kx1	Call
Z-80A	CPU	7.25	Z80B	CPU		21.00

NEC 1Kx4 STATIC RAM 250 N.S.

These are prime low power static ram's NEC for the finest in MOS MĚMORY.

2114L	250ns
8 for $25.00	32 for $96.00

Z80-P10	6.00	8755	49.95	4050	.69	4531	.99	74C925	6.95	74LS107	.45	74LS244	1.95
Z80A-P10	7.10	4000	.35	4051	1.10	4532	1.25	74LS00	.35	74LS109	.45	74LS245	4.95
Z80-CTC	6.00	4001	.35	4052	1.10	4539	.99	74LS01	.28	74LS112	.49	74LS247	1.10
Z80ACTC	7.10	4002	.35	4053	1.10	4543	1.99	74LS02	.28	74LS122	.55	74LS248	1.10
Z80-DMA	18.50	4006	1.39	4055	3.95	4553	3.50	74LS03	.28	74LS123	1.19	74LS249	1.69
Z80A-DMA	22.50	4007	.29	4056	2.95	4555	.75	74LS04	.39	74LS125	1.35	74LS251	1.79
Z80-S10/0	18.50	4008	1.39	4059	9.95	4556	.75	74LS05	.28	74LS126	.89	74LS253	.98
Z80A-S10/0	22.50	4009	.49	4060	1.39	4581	1.99	74LS08	.39	74LS132	.79	74LS257	.98
Z80-S10/1	18.50	4010	.49	4066	.75	4582	1.01	74LS09	.39	74LS136	.59	74LS258	.98
Z80A-S10/1	22.50	4011	.35	4068	.35	4584	.55	74LS10	.28	74LS138	.89	74LS259	2.95
Z80-S10/2	18.50	4012	.29	4069	.35	4585	.99	74LS11	.39	74LS139	.89	74LS260	.69
Z80A-S10/2	22.50	4013	.49	4070	.49	4702	9.95	74LS12	.39	74LS145	1.25	74LS261	2.49
3205	3.95	4014	1.39	4071	.35	74C00	.39	74LS13	.47	74LS148	1.49	74LS266	.59
3242	10.00	4015	1.15	4072	.35	74C02	.39	74LS14	1.25	74LS151	.79	74LS273	1.75
8155	11.25	4016	.59	4073	.35	74C04	.39	74LS15	.39	74LS153	.79	74LS275	4.40
8185	29.95	4017	1.19	4075	.35	74C08	.49	74LS20	.26	74LS155	1.19	74LS279	.59
8185-2	39.95	4018	.99	4076	1.29	74C10	.49	74LS21	.38	74LS156	.99	74LS283	1.10
8202	45.00	4019	.49	4078	.35	74C14	1.65	74LS22	.38	74LS157	.99	74LS290	1.29
8205	3.95	4020	1.19	4081	.35	74C20	.39	74LS26	.39	74LS158	.75	74LS293	1.95
8212	2.00	4021	1.19	4082	.35	74C30	.39	74LS27	.39	74LS160	.98	74LS295	1.10
8214	3.95	4022	1.15	4085	1.95	74C32	.99	74LS28	.39	74LS161	1.15	74LS298	1.29
8216	1.85	4023	.38	4086	.79	74C42	1.85	74LS30	.26	74LS162	.98	74LS324	1.75
8224	2.65	4024	.79	4093	.99	74C48	2.39	74LS32	.39	74LS163	.98	74LS347	1.95
8226	1.85	4025	.38	4099	2.25	74C73	.85	74LS37	.79	74LS164	1.19	74LS348	1.95
8228	5.00	4026	2.50	4104	1.99	74C74	.85	74LS38	.39	74LS165	.89	74LS352	1.65
8238	5.45	4027	.65	4501	.39	74C85	2.49	74LS42	.79	74LS166	2.49	74LS353	1.65
8243	4.65	4028	.85	4502	1.65	74C89	4.95	74LS47	.79	74LS170	1.99	74LS363	1.49
8251A	5.55	4029	1.29	4503	.69	74C90	1.85	74LS48	.79	74LS173	.89	74LS365	.99
8253	9.85	4030	.45	4505	8.95	74C93	1.85	74LS51	.26	74LS174	.99	74LS366	.99
8255A	5.40	4031	3.25	4506	.75	74C95	1.85	74LS54	.35	74LS175	.99	74LS367	.73
8255A-5	5.40	4032	2.15	4507	.95	74C107	1.19	74LS55	.35	74LS181	2.20	74LS368	.73
8257	9.25	4033	2.15	4508	3.95	74C151	2.49	74LS73	.45	74LS190	1.15	74LS373	2.75
8257-5	9.25	4034	3.25	4510	1.39	74C154	3.50	74LS74	.59	74LS191	1.15	74LS374	2.75
8259A	7.30	4035	.95	4511	1.39	74C157	2.10	74LS75	.68	74LS192	.98	74LS375	.69
8271	60.00	4037	1.95	4512	1.39	74C160	2.39	74LS76	.45	74LS193	.98	74LS377	1.95
8275	32.95	4040	1.29	4514	3.95	74C161	2.30	74LS78	.65	74LS194	1.15	74LS385	1.95
8279	10.80	4041	1.25	4515	3.95	74C163	2.39	74LS83	.99	74LS195	.95	74LS386	.65
8279-5	10.80	4042	.95	4516	1.69	74C164	2.39	74LS85	1.19	74LS196	.89	74LS390	1.95
8282	6.70	4043	.85	4519	.99	74C173	2.59	74LS86	.45	74LS197	.89	74LS393	1.95
8283	6.70	4044	.85	4520	1.39	74C174	2.75	74LS90	.75	74LS221	1.49	74LS395	1.70
8284	5.85	4046	1.75	4522	.99	74C175	2.75	74LS92	.75	74LS240	1.95	74LS399	2.95
8286	6.70	4047	1.25	4526	1.15	74C192	2.39	74LS93	.75	74LS241	1.90	74LS424	2.95
8287	6.70	4048	.99	4527	1.75	74C193	2.39	74LS95	.88	74LS242	1.95	74LS668	1.75
8288	25.40	4049	.69	4528	.99	74C195	2.39	74LS96	.98	74LS243	1.95	74LS670	2.29
8289	49.95												

MAIL ORDERS SHOULD BE SENT TO:
P.O. Box 21432 Seattle, Washington 98111
Telephone Orders & Inquiries (206) 453-0792
Minimum Order $10.00 Add $3.00 Shipping

HANLEY ENGINEERING ✓5

RETAIL STORE
1644 116th NORTHEAST
BELLEVUE, WASHINGTON 98005

FOR THE FINEST IN MICROPROCESSOR SUPPORT I.C.'S

that the number of digits required for all intermediate calculations will not exceed N + M, the number of digits needed to express the desired product.

Defining the subscripted values is done using:

```
 70 FOR I=N to 1 STEP -1
 80 READ A(I)
 90 NEXT I
100 FOR I=M to 1 STEP -1
110 READ B(I)
120 NEXT I
```

Now let's examine a procedure that eliminates the need for excessive storage of intermediate results. The new procedure begins to differ with the first multiplication.

Standard procedure

```
     2
 31647
* 423
     1
```

The 2 to be "carried" is recorded in a scratch area.

New procedure

```
 31647
* 423
    21
```

The 2 to be "carried" is recorded in the appropriate position of the intermediate result. Notice that this is the position to which the carry would eventually be added using the standard procedure.

After 31647 is multiplied by 3, the results of both procedures will appear the same, but that is where the similarity ends. After then multiplying 7 by 2, the two procedures yield:

Standard procedure

```
     1
 31647
* 423
 94941
     4
```

The 4 is recorded on a new line and the 1 to be "carried" is recorded in a scratch area.

New procedure

```
 31647
* 423
941081
```

This one isn't so obvious. The intermediate product was derived from

```
  94 941
+    14
  941081
```

Certainly this number never appears using the standard procedure. In fact, since 10 occupies the space of a single digit, the number never appeared in anything in the student's background.

Continuing the new procedure, we multiply 4 by 2 and add the product to the appropriate digit of the intermediate result to produce

```
 31647
* 423
 95881
```

Don't continue until you know why this intermediate product occurs.

After all digits of 31647 have been multiplied by 2, the two procedures appear as:

Standard procedure	New procedure
31647 * 423 94941 63294	31647 * 423 727881

After all digits of 31647 have been multiplied by 4, the two procedures appear as:

Standard procedure	New procedure
31647 * 423 94941 63294 126588	31647 * 423 13386681

Notice that the new procedure has not only conserved storage space, but also the correct product is now known without additional computation. The BASIC commands required to implement this new procedure are:

```
140 C=0
150 FOR I=1 to N
160 P=A(I)*B(K)
170 C=INT (P/10)
180 P=P-C*10
190 S=I+K-1
200 D(S)=D(S)+P
210 C1=INT(D(S)/10)
220 D(S)=D(S)-C1*10
230 D(S+1)=D(S+1)+C+C1
240 NEXT I
250 NEXT K
```

After understanding this algorithm, there remains only the task of printing the final product. Leading zeroes are first eliminated using:

```
260 N=S+1
270 IF N>1 THEN IF D(N)=0 THEN N=N-1 : GOTO
    270
```

The complete program is given in Listing 3. Adding a line 235 that prints the values of all digits in D will help illustrate the intermediate values of the new procedure.

Factorials

The factorial of an integer is defined as the product of all integers 1 through the given integer. For example, 5 factorial is 1*2*3*4*5, which equals 120. Calculating the exact value of large factorials is a natural extension of the program illustrated in Listing 3. The hard part has already been done in lines 130 through 250. We need only change the initial values and the output format to complete a program that calculates virtually any factorial we desire.

Listing 4 contains a program with all of the suggested changes. The program prints the factorials of all integers 1 through 999. Lines 10 through 40 set the initial values, and lines 320 through 390 reset these values after each factorial is computed.

Don't miss the opportunity to share this program with students. Experience suggests that the majority of students enjoy first watching and then experimenting with the large numbers produced. The value of 999 factorial given in Fig. 1 was calculated using a similar program. Can you calculate the exact value of 1000 factorial without pencil, paper or computer?□

```
10 DEFINT A-Z
20 DIM D(1500)
30 N=1 : D(1)=1 : M=3
40 FOR K=1 TO 1690
50 C=0
60 FOR I=1 TO N
70 P=D(I)*M+C
80 C=P/10
90 D(I)=P-C*10
100 NEXT I
110 D(N+1)=C
120 IF C>0 THEN N=N+1
130 PRINT : PRINT : PRINT
140 PRINT "    " K
150 PRINT "3"
160 PRINT
170 FOR I=N TO 1 STEP -1
180 PRINT D(I);
190 NEXT I
200 NEXT K
```

Listing 2.

```
10 'DATA 5
20 DATA 3
30 DATA 3,1,6,4,7
40 DATA 4,2,3
50 READ N, M
60 DIM A(N), B(M), D(N+M+1)
70 FOR I=N TO 1 STEP -1
80 READ A(I)
90 NEXT I
100 FOR I=M TO 1 STEP -1
110 READ B(I)
120 NEXT I
130 FOR K=1 TO M
140 C=0
150 FOR I=1 TO N
160 P=A(I)*B(K)
170 C=INT(P/10)
180 P=P - C*10
190 S=I+K-1
200 D(S)=D(S) + P
210 C1=INT(D(S)/10)
220 D(S)=D(S) - C1*10
230 D(S+1)=D(S+1) + C + C1
240 NEXT I
250 NEXT K
260 N=S+1
270 IF N>1 THEN IF D(N)=0 THEN N=N-1 : GOTO 270
280 PRINT "THE PRODUCT IS:"
290 FOR I=N TO 1 STEP -1
300 PRINT D(I);
310 NEXT I
```

Listing 3.

```
10 DEFINT A-Z
20 DIM A(3000), B(3), D(3000)
30 A(1)=1 : B(1)=1
40 N=1 : M=3
130 FOR K=1 TO M
140 C=0
150 FOR I=1 TO N
160 P=A(I)*B(K)
170 C=INT(P/10)
180 P=P - C*10
190 S=I+K-1
200 D(S)=D(S) + P
210 C1=INT(D(S)/10)
220 D(S)=D(S) - C1*10
230 D(S+1)=D(S+1) + C + C1
240 NEXT I
250 NEXT K
260 N=S+1
270 IF N>1 THEN IF D(N)=0 THEN N=N-1 : GOTO 270
280 PRINT 100*B(3)+10*B(2)+B(1) "FACTORIAL IS:"
290 FOR I=N TO 1 STEP -1
300 PRINT D(I);
310 NEXT I
320 FOR I=1 TO N
330 A(I)=D(I)
340 D(I)=0
350 NEXT I
360 PRINT : PRINT
370 B(1)=B(1)+1 : IF B(1)<10 THEN 130
380 B(1)=0 : B(2)=B(2)+1 : IF B(2)<10 THEN 130
390 B(2)=0 : B(3)=B(3)+1 : IF B(3)<10 THEN 130
```

Listing 4.

```
   402 387260 077093 773543 702433 923003 985719 374864 210714 632543
799910 429938 512398 629020 592044 208486 969404 800479 988610 197196
058631 666872 994808 558901 323829 669944 590997 424504 087073 759918
823627 727188 732519 779505 950995 276120 874975 462497 043601 418278
094646 496291 056393 887437 886487 337119 181045 825783 647849 977012
476632 889835 955735 432513 185323 958463 075557 409114 262417 474349
347553 428646 576611 667797 396668 820291 207379 143853 719588 249808
126867 838374 559731 746136 085379 534524 221586 593201 928090 878297
308431 392844 403281 231558 611036 976801 357304 216168 747609 675871
348312 025478 589320 767169 132448 426236 131412 508780 208000 261683
151027 341827 977704 784635 868170 164365 024153 691398 281264 810213
092761 244896 359928 705114 964975 419909 342221 566832 572080 821333
186116 811553 615836 546984 046708 975602 900950 537616 475847 728421
889679 646244 945160 765353 408198 901385 442487 984959 953319 101723
355556 602139 450399 736280 750137 837615 307127 761926 849034 352625
200015 888535 147331 611702 103968 175921 510907 788019 393178 114194
545257 223865 541461 062892 187960 223838 971476 088506 276862 967146
674697 562911 234082 439208 160153 780889 893964 518263 243671 616762
179168 909779 911903 754031 274622 289988 005195 444414 282012 187361
745992 642956 581746 628302 955570 299024 324153 181617 210465 832036
786906 117260 158783 520751 516284 225540 265170 483304 226143 974286
933061 690897 968482 590125 458327 168226 458066 526769 958652 682272
807075 781391 858178 889652 208164 348344 825993 266043 367660 176999
612831 860788 386150 279465 955131 156552 036093 988180 612138 558600
301435 694527 224206 344631 797460 594682 573103 790084 024432 438465
657245 014402 821885 252470 935190 620929 023136 493273 497565 513958
720559 654228 749774 011413 346962 715422 845862 377387 538230 483865
688976 461927 383814 900140 767310 446640 259899 490222 221765 904339
901886 018566 526485 061799 702356 193897 017860 040811 889729 918311
021171 229845 901641 921068 884387 121855 646124 960798 722908 519296
819372 388642 614839 657382 291123 125024 186649 353143 970137 428531
926649 875337 218940 694281 434118 520158 014123 344828 015051 399694
290153 483077 644569 099073 152433 278288 269864 602789 864321 139083
506217 095002 597389 863554 277196 742822 248757 586765 752344 220207
573630 569498 825087 968928 162753 848863 396909 959826 280956 121450
994871 701244 516461 260379 029309 120889 086942 028510 640182 154399
457156 805941 872748 998094 254742 173582 401063 677404 595741 785160
829230 135358 081840 096996 372524 230560 855903 700624 271243 416909
004153 690105 933983 835777 939410 970027 753472 000000 000000 000000
000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
000000 000000 000000 000000 000000 000000 000000 000000
```

Fig. 1. The value of 999 factorial.

By Eric Maloney

Name Games

Give Your Product The Moniker It Deserves

As a service to the businessmen among our readers, *Microcomputing* has devised a quick and simple (some might even say dirty) method for naming new products and companies.

Table 1 lists 26 prefixes and 26 suffixes for your use. Simply attach any entry in the first column to any entry in the second, and voila! You have yourself an instant name.

The table provides 668 possible combinations (676, less eight such doubles as Datadata and Videovideo). But if you still aren't satisfied, here are some ways to enlarge the pool:

1. If you're naming a product, stick a number at the end of it. Four-digit numbers ending in three zeros are very contemporary. For example, the Electrometrics 2000.
2. For extra pizzazz, try putting a capital letter in front of one of the suffixes, separating them with a hyphen: M-Comp, T-Flex, G-Net. If you don't like hyphens, slashes are becoming de rigueur. (Using both hyphen and slash—R/B-Type, for example—is considered gauche.)
3. To give your company that high-powered, high-tech profile, put one of the following after the name: International, Research, Systems, Group. In fact, if you want to get really macho, use all four.
4. Finally, if you prefer a more informal image, sandwich your choice between your first name and either Shack or Palace: Willie's Microfax Shack, Bob's Plexiplus Palace. The word Emporium is no longer stylish, although some observers think that Barn is about to make a big comeback.

* * * * *

Speaking of names, has anyone noticed that the data communications industry has gone bonkers for the suffix *-net*? A quick survey of a few of the trade journals uncovers such examples as MicroNet, Miconet, Texnet, PETnet, Applenet, Ethernet, Fordnet, Tymnet, Telenet, Primenet, Macomnet, B/C-Net, Mitrenet, and TermiNet. Pretty soon we'll need a Netnet to keep them all straight.

* * * * *

Microcomputing's authors have recently shown an increasing fondness for the word generate. Computerists don't seem to think anymore; they generate ideas. Those with word processors no longer write; they generate copy. For example, a recent manuscript included the line, "New lead material was generated for the Chicago story."

Generate is a perfectly good verb, and can be effective when used in the right place at the right time. But it is one of an increasing number of crossover words that are being applied to both the human and the computer experience. (Others include interface for talk, data bank for memory and input for information.)

I, for one, prefer to maintain the distinction between man and machine, both in thought and in language. As Melvin Maddocks put it in a recent issue of the *Christian Science Monitor:*

"Until a computer falls in love or laughs at a bad pun or weeps at what its 'memory bank' tells it in the middle of the night, we humans should avoid all but the most limited of comparisons in both directions."

* * * * *

Old Words Learning New Tricks Dept.

—From The Interface Group in Framingham, MA, comes a news release with this heading: "Interface Group Pacts with Little Bros." Close, but no cigar—while impacts is a certified verb, pacts is not. Until somebody comes up with a good reason why it should be otherwise, a pact is still a thing, and it must be made, agreed to or in some way acted upon.

This next one comes from an article in *Computer Systems News:* "A new round of entrepreneurial companies entered the arena, while mature companies continued to ramp up production on both new and established products." This is more confusing than the Interface Group example, for while ramp can be used as a verb, none of its dictionary meanings makes any sense here: "to rear as if to spring," "to leap or dash with fury," "to act violently; rage; storm," "to creep up—used esp. of plants."

From the context, one can deduce that ramp up is being used as a synonym for increase. But this leads to another question: why didn't the author simply use the word increase?

Finally, Aaron Contorer of Deerfield, IL, writes to say that he and a number of CBM operators, all hooked up to one disk drive and a couple of printers, are using the verb disking to indicate when one of them is using the disk drive.

"If any two of us try to print or access the disk drive at the same time, the signals will foul each other up, possibly messing up printouts and erasing or changing disks," he says. "To avert disaster, we have invented the word disking. This is a plain enough word, meaning 'to use the disk drive.' It is probably a spin-off from the word printing, and is much simpler than saying 'going to disk' or 'using the disk drive.' "

People often capriciously alter the meaning of a word because they are sloppy or ignorant. Here is a good example of an adaptation being made to meet a specific situation or need. Disk away, gang.

* * * * *

Speaking of disks, is it possible that someday we'll have antiquarian disk dealers and Great Disk Societies? That college students will hit the disks when studying? That criminals will be hauled off to the precinct to be disked? That the Bible will be referred to as the Good Disk? Just asking. . . . □

Prefixes	Suffixes
Compu	data
Tele	soft
Multi	micro
Data	ram
Inter	net
Digi	com
Auto	type
Uni	tech (tek)
Video	mation
Info	writer
Soft	tex
Mega	serve
Astro	graphics
Micro	fax
Opti	tronix
Dyna	tel
Plexi	plus
Ram	con
Com	file
Techno	plex
Con	comp
File	flex
Insta	video
Meta	calc
Electro	link
Alpha	metrics

Table 1.

Compiled by Eric Maloney

Private Schools

Micros Go Catholic in Texas, Omaha

Private Schools Go Micro

Students at Nolan Catholic High School in Fort Worth, TX, will soon be required to take a microcomputer course if they want to graduate.

The computer literacy program, which starts with the graduating class of 1984, is the direct result of a new computer center donated to the school by several anonymous benefactors. The school, with an enrollment of about 750 students, owns 21 TRS-80 microcomputer systems.

The program has been a big hit with the students so far, says Brother Tony Pistone, Nolan's principal. Some 160 students were enrolled in the one-semester course this past spring, and Brother Tony expects some 220 to take it this fall.

"The computer center is very attractive and centralized," he says. "It's different from classrooms. Students see the possibilities, and it opens a lot of creative avenues for them."

The program developed last year with the help of Brother Eugene Meyerpeter, a computer education specialist from St. Louis, MO. "He told us what it would take, and we submitted this proposal to our benefactor," says Brother Tony. The school placed 15 Model I's in the computer center and two in math classes. Model IIs are being used for attendance records and for payroll and accounting.

Brother Tony says that a couple of math teachers taught the computer course this spring. "But next year we hope to have a full-time teacher who will not only teach the course, but will also help our other departments and faculty to use the computers."

Nolan is also using the computers for adult education courses.

"Some of the parents didn't want the kids to get too far ahead of them," Brother Tony says.

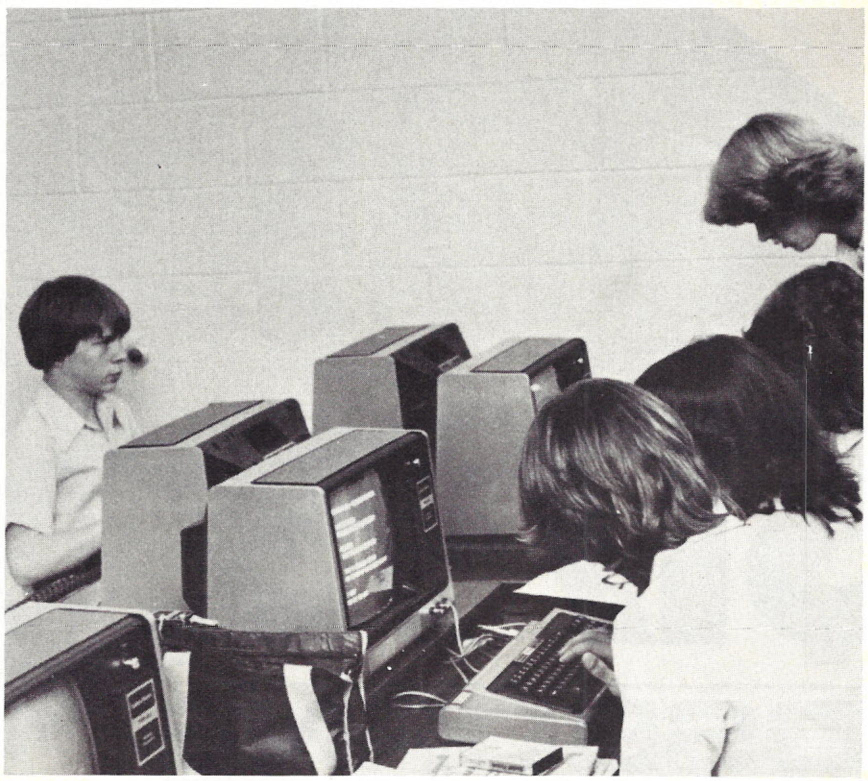
Nolan is not the only Catholic school to be embracing microcomputers. Brother Eugene, who did Nolan's needs assessment, is the new principal at Daniel J. Gross High School in Omaha, NE. That school, with an enrollment of about 1300, will double the number of TRS-80 Model I's it has to 15 or 16, and will use Model IIs for its administrative work.

Brother Eugene is firmly committed to the teaching of microcomputer technology in elementary and secondary schools. "It's vitally important that students be alerted to the importance computers will have in their lives after they graduate," he says.

Daniel J. Gross High School currently has no computer literacy requirement, but Brother Eugene says that he is "certainly going to ask the faculty to consider it." But he cautions that the requirement "must be defined very carefully. I don't think it's feasible to think that all students have, or should have, the same educational needs."

Computer education, he says, "really involves a question of responsibility. It's trite to say this, but it's true: people blame computers for problems that people are responsible for. It should be part of the educational process to change this way of thinking."

Both Brothers Tony and Eugene agreed that teachers are more intimidated by



The computer center at Nolan Catholic High School in Fort Worth, TX, includes 15 TRS-80 Model I's. Starting with the graduating class of 1984, students must take a one-semester course in microcomputers.

the computers than students are.

"Teachers are afraid of them," Brother Tony says. "It's amazing to me, but they are afraid of them."

"There's a whole group of teachers out there who don't even know that this technology exists," says Brother Eugene. "This is what scares me. Business has turned to the computer because of dollars, and has run with it, and it has created a real gap between the business and educational communities."

Are private schools better-equipped to provide the kind of computer literacy Brother Eugene says high school students will need? Probably, he says.

"My bias is that we're much more flexible, because there's less bureaucracy to get around. The schools within each archdiocese are connected, but they are pretty autonomous.

"I've met some very frustrated public school educators who would like to get into it, but can't."

Software Industry Forges (Shakily) Ahead

A recent report from International Resource Development in Norwalk, CT, depicts the software industry as being in constant chaos and flux, but says that it will reach $2 billion in sales by 1985.

The report says that one-person software houses will continue to rise and fall with the tides, and that many venture capitalists will pour millions of dollars into marginal software houses. Thus, says the report, consumers will continue to be the recipients of poor-quality software, until a predicted shakeout around 1983.

The study also says that while the Japanese will take a significant bite out of the hardware market, software will come largely from American manufacturers. The report says that at least two software firms—Friends-Amis and Microsoft—have software contracts with Japanese microcomputer manufacturers. The Japanese will end up with significantly less than half the hardware/software market, says the report, since software will eventually outsell hardware.

The report also concludes that large outfits like IBM, Hewlett-Packard, DEC, Data General and Texas Instruments will dominate the small business software market, while such companies as Radio Shack, Apple and Commodore will scramble for the scraps.

Another report, this one from Venture Development Corporation of Wellesley, MA, says that the business microcomputer market will see an annual growth rate of about 52 percent between now and 1985. The other three major markets —education, engineering/science and home/hobby—will grow at annual rates of 34, 30 and 26 percent, respectively.

The report concludes that total annual shipments of microcomputers will increase from 400,000 in 1980 to 2 million in 1985.

1981

- IBM introduces its System 380, an under-$3000 personal computer with the power of the IBM 370 mainframe.

1982

- Japanese manufacturers launch major invasion of home computer market with powerful personal computers that are compatible with the enormous existing base of Microsoft BASIC and CP/M software, and cost less than $300.
- Knight-Ridder and AT&T announce a nationwide joint viewdata service.
- The United American Service Corp., CompuServe and Radio Shack sell their 500,000th electronic banking-at-home system.
- High-performance, low-cost integrated video terminal for sale.
- Program generators eliminate shortage of programmers, but shortage of capable systems analysts persists.

1983

- All major print publishers either begin software divisions or acquire shares in important software providers.
- All major toy and hobby companies establish software divisions.
- First $100 million-a-year software house.

1985

- The 1000th computer software shop opens.
- Manufacturers agree to standards for systems software and high-level languages.
- English-language programming captures vast majority of software market.

International Resource Development's scenario for microcomputer software, 1981–86.

	No. of Micros Installed (year-end)	Annual Software Unit Sales	Annual Retail Value of Software Sold
1979	250,000	1,000,000	$200 million
1981	750,000	2,150,000	600 million
1983	2,000,000	4,750,000	950 million
1985	4,500,000	10,000,000	2.0 billion
1987	10,125,000	25,300,000	5.1 billion
1990	33,750,000	100,000,000	25.0 billion

Potential sales of microcomputer software through 1990. (Source: International Resource Development, Inc.)

One of England's famed double-decker buses has become a rolling billboard for Datalink Microcomputer Systems Ltd. of Bristol. The design was created by Wedge Wilson Marketing Services Ltd., also of Bristol.

The Insight Series 10

Texas Instruments has entered the videotext terminal market with the Insight Series 10 Personal Information Terminal.

The terminal, which is 12 inches high and has a 5-1/2-inch swivel display screen, is used to access various electronic information services like CompuServe and The Source.

The Series 10, which TI bills as "designed for the nontechnical user," has a built-in modem and plugs into a standard modular wall telephone jack and into the telephone. The user dials a telephone number, types in the log-on sequence and is on-line. A talk/data switch gives the user normal telephone use when required.

The screen has a 24-row by 40-character format, with a wrap-around feature when 80-column transmission is received. An optional command module can store up to eight prerecorded functions, with 32 characters per function.

The Series 10 comes with an EIA RS-232-C interface, and a limited RS-232-C interface for hard-copy printouts.

TI is also marketing an Insight Series 10/1 receive-only thermal printer.

Both the Series 10 and the Series 10/1 are priced at $995.

CompuPro Recall

CompuPro, a division of Godbout Electronics, has issued a general recall of its CPU 8085/88 dual processor board that contains 8085A-2 CPU chips manufactured by AMD. The chips are marked as 5 MHz parts, but are actually only 3 MHz parts. They are marked AM8085A-2CC (ceramic package) and AM8085A-2PC (plastic package).

Boards that contain 8085s from Intel are not subject to recall.

Customers who bought their boards assembled and tested should return the board and the AMD part will be replaced. If the board was bought unassembled, you should return the part for a replacement.

Owners of CSC boards should already have received replacement boards.

For assistance or information, contact CompuPro, PO Box 2355, Bldg. 727, Oakland Airport, CA 94614 (415-562-0636).

Micros to China

The People's Republic of China is buying up to $5 million worth of Z-80 based small business microcomputers from Zilog, Inc., of Cupertino, CA.

The Beijing Automation Technology Research Institute will be using MCZ 1/50 microcomputer systems for airline and hotel reservation systems and for educational purposes at universities.

Zilog is training several People's Republic representatives to use the computers and their software.□

✓See List of Advertisers on page 210

LETTERS TO THE EDITOR

Pocket Computer Benchmarks

I noted with interest Nat Wadsworth's article on the TRS-80 Pocket Computer (PC) in the February 1981 issue of *Microcomputing* (p. 162). I recently purchased in Japan a Sharp PC-1211, which is the same as the TRS-80 PC except the PC-1211 has a yen sign (¥) over the Y key.

Wadsworth used the benchmark timing comparisons from the June 1977 issue of your magazine ("BASIC Timing Comparisons, p. 66) to rank the PC. I submit this is like comparing apples to oranges, and thus serves as a limited yardstick for comparison.

Perhaps a better comparison could be made between the PC and Hewlett-Packard's HP-41C or the Texas Instruments TI-59. These three instruments have similar amounts of memory, speed, program storage ability, options and price tags. (The PC I bought can be outfitted with a cassette interface/printer that plugs into the side of the PC and is only slightly larger than the PC itself.)

Some highlights of the features of the PC include the ability to read and write both programs and data through the cassette interface. The PC has the ability to use labels instead of line numbers for program transfer; it supports nested subroutines, formatted output, dimensioned variables and string variables; and it doesn't forget when the power is turned off.

In a recent article in *Byte* (December 1980, p. 244), the comparison was made by Bruce D. Carbrey between the HP-41C and the TI-59 on two different benchmark programs. I would like to include my findings on how the PC measured up to the first benchmark so your readers can judge among the three machines. (See Table 1.)

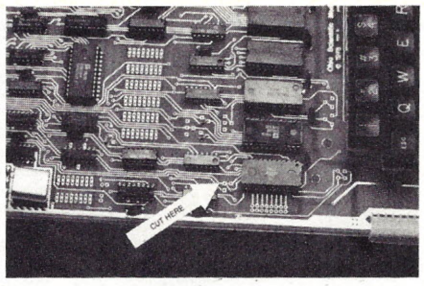


Photo 1.

Note that the PC-1211 doesn't walk away from the competition. When you consider, though, that the programs are written in BASIC, you have full editing and debugging ability and there are many features and options available that contribute to its ease of use and programming, then you have a strong inducement to consider the PC before buying a machine that falls into the category between a microcomputer and a programmable calculator.

Scott U. Johnson
San Francisco, CA

A Simpler Modification

Jim Antonelli in his C1P speedup article ("Faster Baud Rate for the Superboard II," March 1981, p. 112) did a good job on the article, and the modification is a good one. It works! However, it's not as simple as it could be.

In Fig. 1, pin 9 of the U63 is the normal TX clock, which feeds the 6850 ACIA to give the nominal 300 baud cassette interface. U63 is a D flip-flop, so the signal at pin 11 is twice the frequency of the clock at pin 9.

The simplest 600 baud modification involves making only one printed circuit cut (see Photo 1) above the common connection at W5 just above the 6850. This breaks the clock line to the ACIA. Now wire in an SPDT switch so the common terminal connects to the ACIA clock and the two positions select either the normal clock from U63, pin 9, or the 2X clock from pin 11.

Since the tones being generated using this technique are the same as the original, there is no need for the extra work to modify the receive side of the cassette interface. The venerable Kansas City standard used by OSI has so much redundancy that having only half as many cycles of each tone makes virtually no difference in the effectiveness of the interface.

I've been using this for almost two years, and when I have a problem loading a tape it means it's time to clean the tape machine.

Curtis A. Preston
Shalimar, FL

More on Zero Numbering

In commenting on zero numbering (Letters to the Editor, April 1981, p. 29), William McGarry uses an example of using disk track numbers as indices into a table in claiming that "if the first track was track #1 instead of track #0, then either one would have to be subtracted from the track number before indexing, or the first entry in DDATA would have to be filled with dummy data to obtain the proper entry . . . this would mean lost memory, speed and efficiency."

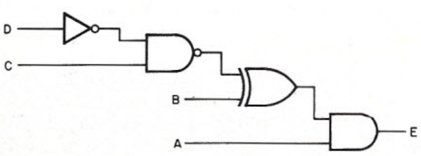
This is indeed true if the subtraction is done at execution time with this code sequence: load track number, subtract one, add address of DDATA, fetch from memory.

If, however, you allow the assembler to do the subtraction, there is no loss in exe-

(continued on page 211)

MICRO QUIZ

Find all combinations of binary inputs (A,B,C,D) which make E true.



(continued on page 184)

Problem: Calculate bond yield using this formula:

$$P = I \sum_{J=1}^{N} (1+Y)^{-J} + M(1+Y)^{-N}$$

Data: M = 20,000, I = 1400, N = 50, Y = 8%
Answer: P = 17,533.30

Results:

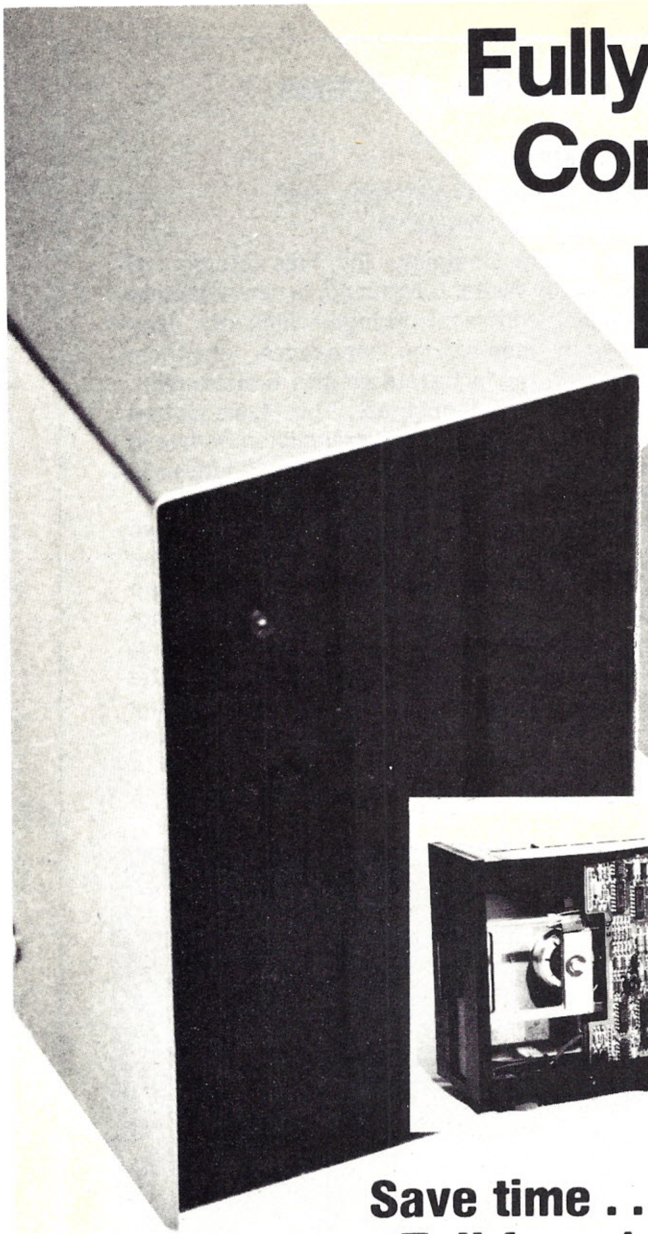
	Time (sec)	Memory used	% of total memory	# of lines
TI-59	43	59 steps	10.3	59
HP-41C	37	41 bytes	15.6	31
PC	51	83 steps	5.8	7

Table 1.

Fully TRS-80® Compatible

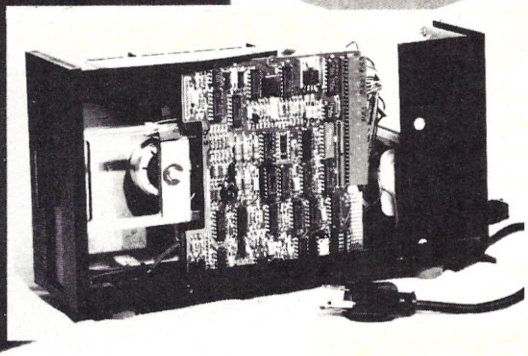
MPI B/51 DISK DRIVE

FREE CATALOG
Hundreds of other computer products at major savings. Cut out this coupon and send with your name and address.



$321.00
We pay UPS charges in Continental US

In Stock! Limited Quantities.



Includes: Case and Power Supply

Fully tested
Guaranteed for 90 days!

Here's why the MPI/B-51 is the drive for success:

- 40 tracks
- 5 ms track-to-track
- Auto-eject
- Hi-Temp stability
- Fully-closable door
- Speed constant <1½%
- Double density head
- Optical sensors—no switches
- 102K per disk

ADDS MORE POWER TO YOUR SYSTEM

Save time . . . Order by phone
Toll free: 1-800-621-3229
IN ILLINOIS CALL: 312-987-1024

TRS-80™ TANDY CORP.

Other Money Savings Opportunities Order by Phone or Mail

CABLES
- 2 Drive Cable........**$29.00**
- 4 Drive Cable........**$39.00**

Diskettes
- Verbatim............10 for **$31.00**
- Dysan ....10 for **$41.00**
- Plastic File Box.......**$3.95**

Operating Systems
- TRSDOS 2.3.........**$14.95**
- 40 Track Patch.......**$9.95**
- NEWDOS + 40 track..........**$99.00**
- Newdos 80........**$139.00**
- TRSDOS Manual......**$5.95**

TRS-80
- 16K Model III **$899.00**
- Expansion Interface ØK............**$274.00**

Printers
- Centronics 737-1....**$815.00**
- IDS 460...........**$1219.00**
- Epson MX 80 ......**$550.00**
- Okidata M80 ......**$679.00**
  - Microline 82 .....**$862.00**
  - Microline 83 ....**$1099.00**

Okidata prices include tractors

- NEC Spinwriter 5530 (freight collect) **$2579.00**

16K Memory Kits
- Prime NEC 200ns dynamic RAM. Comes with complete instructions.........**$39.00**
- MPI Service Manual...........**$3.00**
- MPI Engineering Manual..........**$30.00**

MIDWEST COMPUTER PERIPHERALS
1467 S. Michigan Ave. Chicago, IL 60605

030181

Midwest Computer Peripherals
1467 S. MICHIGAN • CHICAGO, ILLINOIS 60605

✓150

master charge · VISA · AMERICAN EXPRESS Card
ACCEPTED!

030181

Quantity	Description	$ each	Total
		6% Ill. Tax	
		TOTAL	

(Min. Order $10.00)

☐ Check enclosed
Bill my ☐ Visa ☐ Am Ex
☐ Master Charge
Acc. No. ____________ Exp. ______
☐ Please send catalog

Name ____________
Address ____________
City ____________ State ______ Zip ______

ENGINEERING AND PRODUCTION CONTROL SYSTEMS

Company founder and president, Walter L. Myers (right), pictured with production engineer Joe Zellers who developed the production control software for the MSI computer system.

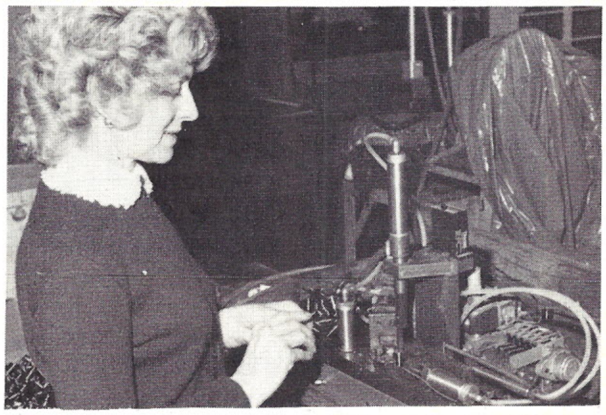
THE COMPANY:

MYERS SPRING COMPANY, INC. LOGANSPORT, INDIANA

Myers Spring Company, Inc. was founded 35 years ago by Walter L. Myers for the manufacture of small mechanical springs which are used widely in mechanical appliances, electrical equipment, and by the automotive, construction, and many other industries. The Myers Spring company has grown to several million dollars in annual sales and employs approximately 50 people in its production facility.

Production engineer Joe Zellers comments, "we began looking at computer systems approximately ten years ago in order to keep up with the increasing demand of order processing, custom mechanical spring design engineering, and production control. In 1975, we selected the MSI system because they were the first company in the microcomputer industry to offer the necessary peripherals which would convert a microcomputer system into a usable business system. Since 1975, the MSI system has been expanded to accommodate four users simultaneously, performing a variety of plant monitoring functions and management programs."

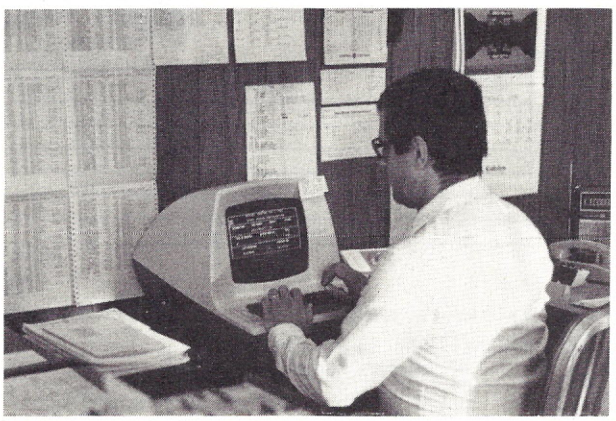
"The system is equipped with 10 megabytes of hard disk storage presently and we plan to add an additional 10 megabytes of storage soon. The maximum downtime has been only one or two days in the 5 years that the system has been in operation. MSI has provided excellent technical support and willingness to help us with our special requirements. We have nothing but praise for MSI, they have provided excellent system support."



The production facility at Myers Spring Co. is equipped with many automated machines for mechanical spring production.

SPRING ENGINEERING AND DESIGN

"All of the production at Myers Spring is performed to exact customer specifications rather than to the manufacture of standard spring products. This causes an ever increasing demand for quick and efficient design and engineering capabilities. Many parameters have to be taken into consideration in the design of any particular spring, including wire size, wire type, material modulus, spring diameter, number of turns per inch, free length, spring loading, rate, solid height, working stress, working temperature, number of operating cycles, hysteresis, resonant frequency, expansion, and whether the spring has to be ground or not. It used to take over an hour for an engineer to design a spring taking into account all of these parameters. However, with the engineering software which we have developed for the MSI system, spring design can be completed in less than one minute by simply keying in the desired parameters. The MSI computer system not only designs the spring for us but prepares a complete quotation for the customer after consideration of the material to be used, the amount of waste, which equipment the production will use, the speed of the machines, the necessary labor rate, as well as the desired percentage of profit."



One of four workstations where design engineering, checking of sales order status, and production control monitoring is performed.

SALES QUOTATION SYSTEM

Following the computer spring design procedure, with automatic quotation feature, the actual production begins. Each quotation is reviewed and compared to actual job cost reports on the production run in order to make any necessary refinements in the quotation system software. This feature of our system has greatly improved our ability to prepare accurate quotations and to insure profitability of the company.

PRODUCTION CONTROL/JOB COST ACCOUNTING

Each production work order is tracked by the computer system at each stage of the production process. First, each order is checked against the customer quotation for accuracy. As each order is processed, exact shop labor time is recorded, for each production machine used, and each stage of the production process. Summary reports are produced showing the total amount of material used, time used on each production machine, amount of material used, and a total cost figure for each work order.

SALES SUMMARY REPORTS

The system is designed to produce monthly sales summaries which show the amount of products sold by each salesman, complete with dates, order numbers, type of product, quantity, type of material, material cost, sales commissions, etc. Totals for each desired category and for each salesman are reproduced.

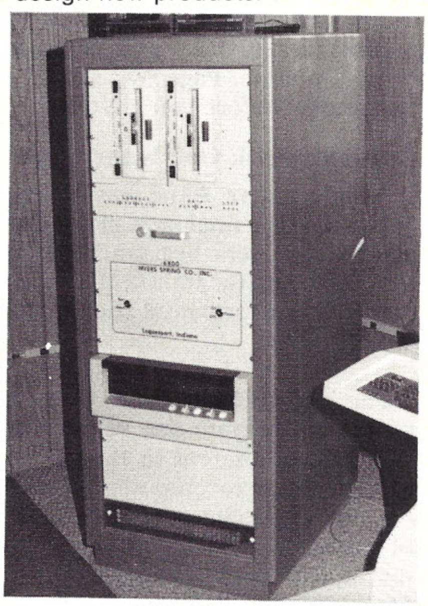
ACCOUNTS RECEIVABLE SYSTEM

Each morning, invoices are generated for orders which will be shipped that day. The accounts receivable system maintains accounts for over 500 active customers. The system produces monthly statements complete with aging of open invoices.

MULTI USER CAPABILITY

The MSI system is equipped with four user terminals presently which are available for use simultaneously by the following departments: Order department, for entering new orders and checking order status. Inventory department, used for checking to see whether a particular product has been produced previously. Invoicing/Cost Accounting, used for preparation of invoices and for entry of labor and material cost accounting information. Design Engineering, used by company engineers to design new products.

Order entry, invoicing, monthly statements, and other management reports are carried out at this workstation at Myers Spring Co.



The MSI system at Myers Spring Company is equipped with 10 megabytes of hard disk memory, dual floppy disk drives, a high speed printer, and four user CRT terminal workstations.

GENERAL LEDGER TIE-INS

The MSI system automatically prepares journals for cost accounting information and sales data which can then be posted to the general ledger. Complete income statements and balance sheets are produced by the general ledger programs on the system.

MULTIPLE MANAGEMENT REPORTS

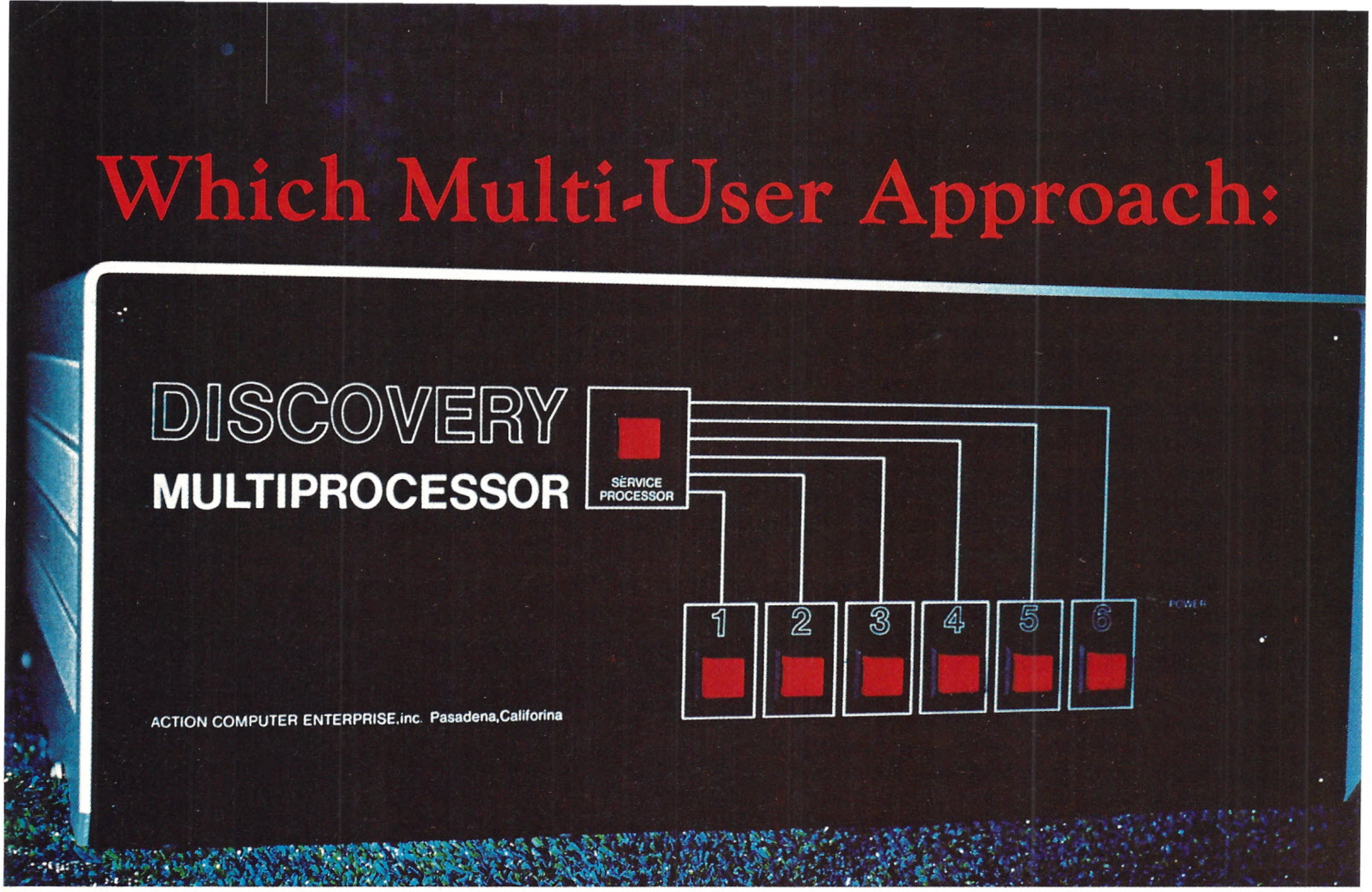
The MSI system is used in many different areas of the company in order to provide more efficient and effective management of our production facilities. Several of the reports which we obtain from the system are: **production schedules, due list for orders, new orders list, production summary by department, salesman's reports, individual customer reports and order histories, time studies, sales quotations, design engineering, sales summaries, customer statements, general ledger balance sheet and income statements.**

Consider MSI, evaluate us, talk to our users, and we think you'll buy your business system from us!

If you would like to know how an MSI business system can help you make your business more profitable, call or write, **MIDWEST SCIENTIFIC INSTRUMENTS, INC., 220 W. Cedar, Olathe, KS 66061, 800-255-6638, Telex 42525 or 437049**

MSI ...... Helping to make your business run better.

Which Multi-User Approach:



The Discovery Multiprocessor from Action Computer Enterprise. S-100 bus and CP/M compatible, this computer supports six user terminals and one printer. Each user has his own CPU and memory, with an additional service CPU providing communications and peripheral handling.

Act I Scene i

It is late at night in the offices of Cribble, Freres, & Brothers, attorneys-at-law. An overtime crew is working furiously to prepare the 14 pages of text that compose a pleading in the case of Grimmich vs Crimwaddy. The document must be perfectly phrased, flawlessly typed and in the hands of Judge Meanie of the District Circuit Court of Dismissals first thing the next morning, if their client's case (or maybe even their client) is to survive.

Anne Attorney sits at her desktop word processing computer, with its whirring fan and buzzing floppy disk drives. She has spent hours composing the text of the pleading. As she completes the last page, she turns to Joe Steno, who sits, bored, at his text formatting/printing computer across the room.

"Joe, here's the pleading. Patch in the standard header, get all the names consistent, and type a rough draft." Anne removes the disk containing the text and holds it out toward Joe.

Joe shuffles across the lavishly carpeted office and reaches for the disk. As their hands come close, molecules are ionized and a static discharge arcs from Joe's hand to Anne's. The static finds a path to ground through Anne's other hand, which is resting on the keyboard of her computer. On its way, the discharge converts all of the contents of memory in her computer to "00 39."

The spark also causes a reflex action in Joe's arm, and he drops the flexible disk onto Anne's overflowing ashtray. Mylar and millions of bits of data melt on the glowing tip of her burning cigarette.

All is lost.

Act II Scene i: Another Time

ENTER: The multi-user microcomputer

Anne sits calmly at her silent terminal, composing the pleading in the continuation of the Grim. vs Crim. case. As she completes the first page of text, she presses the button on her intercom.

"Joe," she whispers seductively, "the first page of the text is on the hard disk as 'GRIM-1.TXT.' Patch in header format 3 and replace all the occurrences of '*G*' with the plaintiff's full name. Page two should be on the disk before you are done."

Act II Scene ii: One Hour Later

Minutes after Anne has finished composing the final page of the document, Joe enters her sanctum with her coffee and the final draft of the 14 pages, neatly formatted and typed by the computer.

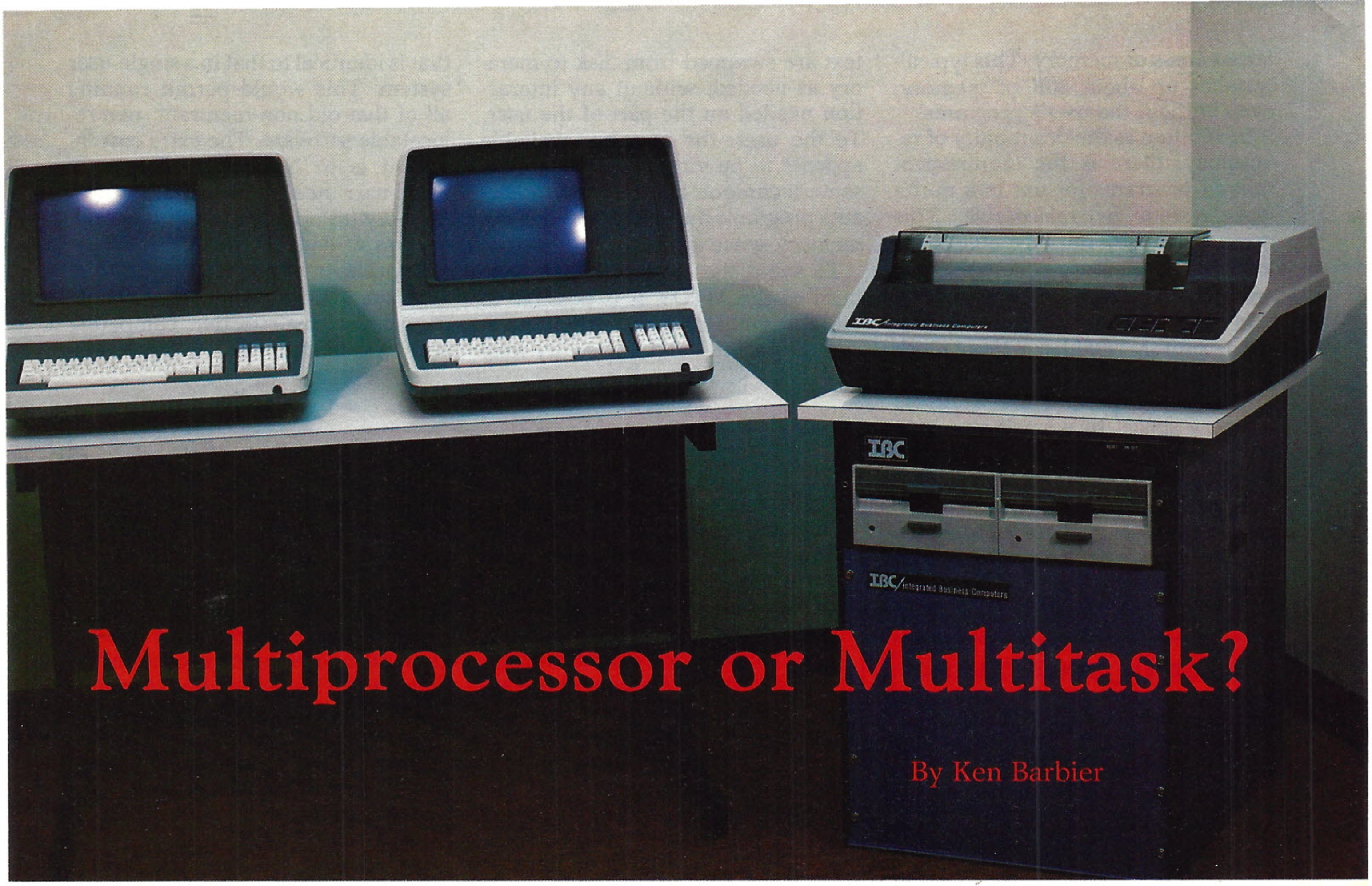
* * * * *

The preceding fictional nonsense serves to illustrate one of the many advantages inherent in the use of a multi-user microcomputer. As microcomputers proliferate in the worlds of business, law and science (to name just a few applications), more and more instances occur in which it would be desirable to have more than one user working with the same database at the same time.

Ken Barbier (Borrego Engineering, PO Box 1253, Borrego Springs, CA 92004) is a Microcomputing *associate editor. His most recent article is "A Print Utility for CP/M," in the February issue.*

Multiprocessor or Multitask?

By Ken Barbier



The System 40 from Integrated Business Computers. While shown with two CRT terminals and one printer, the IBC computer supports four user terminals and two printers at the same time. The basic computer comes with two double-density, double-sided floppy-disk drives. A 20 Mbyte hard disk can be added.

This mode of operation has traditionally been the province of the large mainframe computer with its megabuck price tag. But as hardware prices plummet, it becomes economically feasible for even the smaller businesses or law offices to own a multi-user microcomputer suitable for accounting, word processing, inventory or production control.

But just what is a ''multi-user'' computer?

Some Definitions

A computer which can support multiple users working independently at multiple terminals, each having complete access to all of the resources of the machine, is defined as a *multi-user* computer (terms in italics are further defined in the Glossary). If a computer has multiple terminals attached, but the terminals can only be used for dedicated applications, such as logging production line output or parts usage, it is a *multiterminal* computer, which is obviously not the same.

Several techniques in both hardware and software have evolved to support a multi-user environment. The traditional method, *multitasking*, uses a single central processing unit (CPU) to handle the programs for all of the users. A pulse generator causes periodic interrupts to occur, and at each interrupt the *task scheduler* stops the program in progress, saves its context and enables the next task, and therefore the next user, to run his program on the CPU. Each user in turn is given one interrupt-defined *time slice* in which to use all of the resources of the CPU. At the end of the current user's time slice, he is put on hold by the scheduler, and the next user gets to run his program.

Since each time slice is only a few ms, it appears to the users that they are all running simultaneously. This word is often misused in reference to multitask systems. Tasks do not run at the same time; they only appear so to us slow humans. Tasks actually execute sequentially in such a system.

For the users to run their programs simultaneously, a *multiprocessor* computer is required, where each user has his own CPU hardware, and only shares such resources as mass storage and line printers. Since each user has full-time access to his own CPU, this method requires one additional control CPU running a supervisory or executive program to handle the shared resources.

Which Method Is Best?

A multitask system makes most efficient use of hardware resources. Memory space can be dynamically allocated to each user according to his needs. If the operating system is properly written, system programs such as an assembler, BASIC or FORTRAN languages, or a word processor are *reentrant*. This means that more than one user can run the same system program at the same location in memory, sharing the program in time slices just as they share the CPU.

Non-reentrant programs require a separate memory image of the program for each user, and this can eat up a lot of memory space in a hurry. For instance, if four users each want to program in BASIC, a non-reentrant version of a BASIC interpreter has to be replicated four times in four dif-

ferent areas of memory. This typically takes up about 80K of memory, even without the user's programs!

In addition to the desirability of reentrancy, there is the requirement that all programs for use in a multitask system be *relocatable*. This means that they must be able to be loaded into memory at any address and still execute properly.

In either case, the multitask operating system (MTOS) has a lot of work to do, and the MTOS itself will require a large part of the available memory space. In a small system it will be almost imperative for the MTOS to support *virtual storage* (VS). A simple example of VS is incorporated in most word processor programs. All of the text in a document file is not in memory at any one time. Blocks of text are swapped from disk to memory as needed, without any interaction needed on the part of the user. To the user, the memory available appears to be virtually infinite. The same technique can be used to execute programs that are larger than the memory space available.

To avoid the large software overhead of the MTOS, it is possible to build a multiprocessor where each user has a CPU and a memory map that is identical to that in a single-user system. This would permit running all of that old non-reentrant, non-relocatable software. The extra cost involved is in hardware, since now each user needs his own physical memory (as opposed to virtual memory), as well as his own CPU.

As prices of integrated circuits (ICs) continue to drop, multiprocessors become more cost effective, competing with the traditional multitask sys-

Glossary

Not every term in the computer industry (or any other!) is so cast in concrete that a single definition can explain all usages or satisfy all users. The definitions given here apply to the types of computer systems covered in this article. Alternate definitions can sometimes apply in other contexts. For instance, some multiprocessors have many CPUs working on the same problem at the same time.

Batch processing—The execution of programs one at a time, where the first program runs to completion before the second can start.

Multiprocessor—A computer system providing each user with his own central processing unit (CPU) and memory. Mass storage (tape or disk drives) and peripherals such as line printers are shared by all users, and are controlled by an additional CPU, the service processor.

Multitask (also multiprogram)—An operating system permitting more than one program to run on a single CPU. Each program in turn is given a small time-slice in which to execute, giving the appearance to the user that all tasks are executing at the same time.

Multiterminal—A computer to which more than one terminal device can be connected. The additional terminals do not have full computational capabilities as is true for a multi-user system.

Multi-user—A computer permitting more than one user to have access to all of the resources of the system at the same time.

Print spooling—The technique of automatically outputting files to the printing device without user intervention. Once print requests are specified, the users can proceed with other tasks while the spooler program handles the slow printing.

Reentrant—Programs or program segments written to permit their use by more than one task at a time. Each task must supply its own data area in memory, and provide the reentrant routine with a register pointing to that area, so that an interrupting task will not modify another task's data.

Relocatable—Programs that, after being assembled or compiled, are in a format that permits their being loaded into memory and executed at any memory address.

Service processor—In a multiprocessor computer, the additional CPU which controls interprocessor communications and the common peripherals.

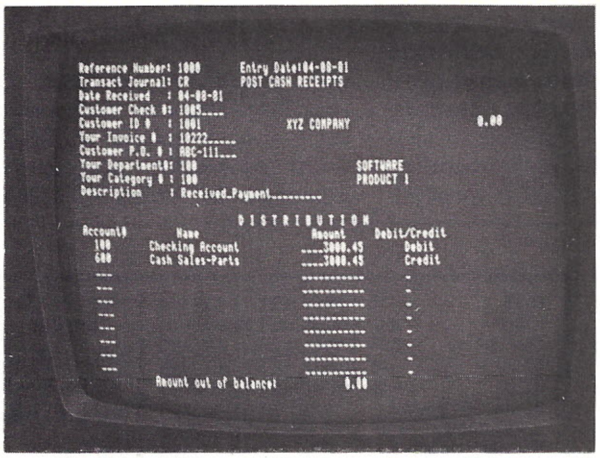
Task scheduler—The program segment within a multitask operating system that provides for the orderly interruption of the current task, the saving of its context (register contents) and the activation of the following task.

Time slice—The increment of time during which each task is allowed to run. Time slices are short enough that each user appears to have full use of the computer all the time.

Virtual storage (also virtual memory)—The operating system technique that permits programs or data to exceed the physical size of main memory. Program or data segments are swapped from disk to memory as needed, with the swapping being invisible to the users, who need not be concerned with the physical size of memory.

Computerize your bookkeeping without terrifying your bookkeeper.

Introducing The Boss:™ the most advanced, yet most understandable, financial accounting system. Designed to automate your bookkeeping without confusion or mistakes.



Typical screen format - actual photograph.

If "fear of the unknown" is standing between you and computerization, you should find out about The Boss.

The Boss system is immediately comprehensible to bookkeepers because it utilizes virtually the same format they're used to.

So even someone with no previous computer experience can easily learn and operate The Boss system.

Features seldom found in packaged software.

The Boss system is fully interactive, fully departmentalized and exceptionally fast. It can generate an astounding number of complex reports at the touch of a button.

General Ledger and Accounts Receivable and Payable transactions can be entered in any order in a single program.

Up-to-the-minute financial reports can be obtained without batch processing.

The Boss system computes financial ratio analysis.

It protects data from unauthorized personnel as well as computer malfunction.

And it has the largest programming and storage capacity of any micro system.

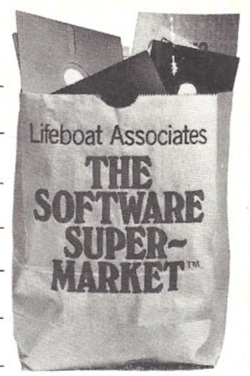
The Boss runs on most small business computers with CP/M® or similar operating systems. Its cost is only $2,495.

Get full support from Lifeboat.

The Boss is brought to you exclusively and supported completely by Lifeboat Associates, world's largest computer software publisher. For more information about how you can profit from this extraordinary financial accounting system, send us the coupon below.

Or call (212) 860-0300

For more information on The Boss, mail coupon to Lifeboat Associates, 1651 Third Avenue, New York, NY 10028.



Name________________

Title________________

Company________________

Street________________

City________________

State________________

Zip________________

BKM681

Lifeboat Associates

Software with full support

✓211

tems. To get a feel for the good and bad points of each of these approaches, let's look at two examples of multi-user microcomputers currently available. The first is a multitask system; the second a multiprocessor.

These two examples have been chosen because they are both fairly new systems, are very affordable, and have been designed from the ground up for multi-users. They are not just adaptations of older hardware and software. One result of starting with a clean slate is that there are no constraints on the designers of either the hardware or software, and the resulting product should be the best possible solution to the multi-user problem.

Either system obviously can be operated as a single-user computer, but their true value comes to light when a second work station is attached. In this two-terminal configuration, either system can be purchased for under $10,000, including software. Adding more work stations to the multiprocessor will cost more than similar additions to the multitask system, as each user needs his own CPU and memory as well as a terminal. Other cost and convenience tradeoffs will be discussed later.

IBC System 40

The System 40 from Integrated Business Computers (22010 S. Wilmington Ave., Carson, CA 90745) is truly integrated, in that the complete computer with a Z-80A microprocessor running at 4 MHz, 128K of RAM, six serial I/O ports and both floppy disk and Winchester hard-disk controllers are all included on a single 15-inch-square printed circuit (PC) board. This board is mounted along with power supplies and two double-sided dual-density floppy-disk drives in a seven-inch-high rack-mountable enclosure. Simply plug any combination of CRT terminals and printers into the six RS-232 connectors on the rear of the computer and you're in business.

The base system price includes 64K of RAM, the pedestal enclosure shown under the printer in the photograph, and a choice of one operating system: either MVT FAMOS or OASIS from Phase One. Both operating systems include the usual array of editor, assembler, debugger and file handling programs. Both also support their own versions of BASIC and word processing programs. Other languages, including FORTRAN, CO-

The Sources

IBC/Integrated Business Computers
22010 S. Wilmington Avenue
Suite 306
Carson, CA 90745
213-518-4245

Action Computer Enterprise, Inc.
55 W. Del Mar Blvd.
Pasadena, CA 91105
213-793-2440

MVT Microcomputer Systems, Inc.
9241 Reseda Blvd., Suite 203
Northridge, CA 91324
213-349-9076

Phase One Systems, Inc.
7700 Edgewater Drive, Suite 830
Oakland, CA 94621
415-562-8085

BOL and Pascal, are either available or are being developed, so the potential purchaser should check with IBC to determine current availabilities and prices.

The IBC computer has hardware features designed to increase reliability and reduce servicing costs. With all the circuitry on a single board, inexperienced service people could easily provide field repairs. If a quick check of power supply voltages shows all present, the trouble is isolated to either the single board or the disk drives, and a quick swap can be made.

Reliability is enhanced because the System 40 includes a minimum of connectors. Peripherals attach to the PC board with ribbon cables, and there is no motherboard or back plane with a multitude of card edge connectors. Total circuitry is minimized as well, as no ICs are needed to implement an expandable bus structure. An additional benefit is reduced power consumption.

The disadvantage of this busless approach is that there is no provision for adding peripherals other than those supported by the basic machine. If some other device proves to be an absolute necessity, one of the six RS-232 ports will have to be dedicated to this use. Peripherals currently available from other vendors, which can be connected through an RS-232 port, include battery operated calendar clocks, modems, analog-to-digital converters, industrial controllers and even PROM programmers.

With only six ports available for user terminals, printers and these other options, expandability is limited. But in the business environment for which the IBC System 40 was designed, the lack of an expansion bus should prove no problem.

The Discovery from ACE

Action Computer Enterprise, Inc., produces the Discovery Multiprocessor, also Z-80A-based, but with a separate processor for each user. The Discovery provides a friendly environment for present microcomputer users, as it is compatible with S-100 bus hardware and CP/M software.

Each user terminal connects via an RS-232 interface to its own Z-80A CPU with 64K of RAM. One additional processor controls the S-100 bus and provides access to floppy- and hard-disk drives. This service processor can also provide access to any other S-100-compatible peripheral, with the only constraint being that all plug-in controllers have to be I/O-mapped devices, rather than memory-mapped. A user of an existing S-100-based system can therefore salvage a lot of hardware, and plug it into the Discovery.

Learning a new operating system will not be necessary for current users of CP/M. Discovery's service processor runs the Distributed Processing Operating System (DPOS), which will load any user processor with any version of CP/M. To demonstrate the flexibility of this multiprocessor, two users have simultaneously run two different versions (1.4 and 2.2) of CP/M, sharing disk accesses on both floppy- and hard-disk drives.

It is this familiar hardware and software environment that is Discovery's strong point. All CP/M-compatible software can run on the system, and since each user has his own 64K memory space, there is no need for relocatable programs. Discovery can even be comfortably used for assem-

bly-language, or even machine-language, program development and debugging, as the service processor and DPOS ensure that each user runs without interfering with his neighbors.

Each user even has his own hardware reset switch, in case he gets into an infinite loop, or executes a HALT instruction. And yet each user can still safely access all of the common resources, such as the disks, printers and any other S-100-compatible peripheral plugged into Discovery's 22-slot mainframe. These accesses will have to be made through driver programs installed on the service processor, since user CPUs do not have access to the S-100 bus except through service calls. This provides the hardware and software protection for the other users.

Since the service processor handles all the heavy peripheral tasks, the user CPUs are not burdened with any extra software overhead. Since the resident portion of CP/M takes up only 6K bytes, the user has access to all the rest of his 64K memory space. With the service processor providing features such as print spooling, the user memory space is not burdened with any extra software overhead.

The Discovery can be supplied in a bare-bones configuration, in case the purchaser already owns compatible disk drives, controllers, terminals or printers. This provides a minimum cost upgrade from an existing S-100 based single-user system. And following such an upgrade, all existing CP/M software will be instantly usable on Discovery.

Features Common to Both Computers

A number of features of multi-user systems are included in both computers. Perhaps the most important is in the protection of data files. Since either system will support terminals in different rooms, it is necessary, for example, to prevent the shop foreman from gaining access to the accountant's data and programs.

Users on either computer can specify various levels of access protection for their program and data files. A file can be made available to all other users for read or write, or it can be made read-only, or it can be fully protected, with only the authorized user even knowing it exists.

Another handy feature is print spooling, permitting a user to request the printing of data while he goes ahead with the next job. Since printers are the slowest peripherals on a computer, a system would not be very efficient if it tied up a CPU full-time just to output data to a printer. In the IBC system, printing is just another task handled by the scheduler. In the ACE computer it is a program resident in the service processor.

Winchester hard-disk subsystems are now available for little more than a floppy-disk system cost just five years ago. Floppies have since increased in storage capacity and have dropped in price, but even with the increased storage a floppy-based computer will soon run out of mass storage space as more and more users are added. To support the inevitable, ACE and IBC both supply hard-disk add-ons, with similar (but not identical) capacities and prices.

The IBC System 40 is typically configured with four CRT terminals and two printers on its six serial ports, while the ACE Discovery can have six user processors, and one printer connected to the service processor. Both companies plan expansion to more users at a later date.

Conclusions

The IBC System 40 provides more capability with fewer components, thanks to the single board construction. Reliability, ease of maintenance and low power consumption result. On the negative side, the multitask operating systems exact a heavy toll, requiring from 32K to 48K of memory for the MTOS alone. This means that each user can have less memory allocated to him at any one time, requiring more frequent disk accesses. With only a single Z-80A to service four CRT terminals, two printers, and both floppy- and hard-disk systems, it is inevitable that some tasks

The IBC Ensign Multi-user

IBC/Integrated Business Computers has recently introduced a powerful Z-80-based computer to its multi-user line. The Ensign computer offers an upgrade path to both dealers and end users who need a larger, more powerful system yet want to keep their library of Z-80-based software intact.

Features include:

- Ability to support up to 16 CRT/printers.
- 64K to 768K byte memory, bank selectable at any size from 4K to 64K/bank.
- 6 MHz Z-80B CPU.
- 14, 42, 70 and 150 megabyte Winchester disk drives.
- Both 17 megabyte tape cartridge and nine-track reel-to-reel tape drive.
- Slave twin Z-80s to handle all serial I/O with 2K byte buffer/serial I/O port (32K byte total). Slave Z-80s prevent high serial I/O usage from degrading main CPU throughput.
- A third slave Z-80 to handle all disk/tape I/O and an advanced DMA controller that allows data transfer to any memory whether selected by main CPU or not. This allows very heavy disk I/O without degrading main CPU throughput.

All of these features are on IBC's new segmented single board computer (SSBC), which allows memory and disk speeds not attainable on an S-100 bus.

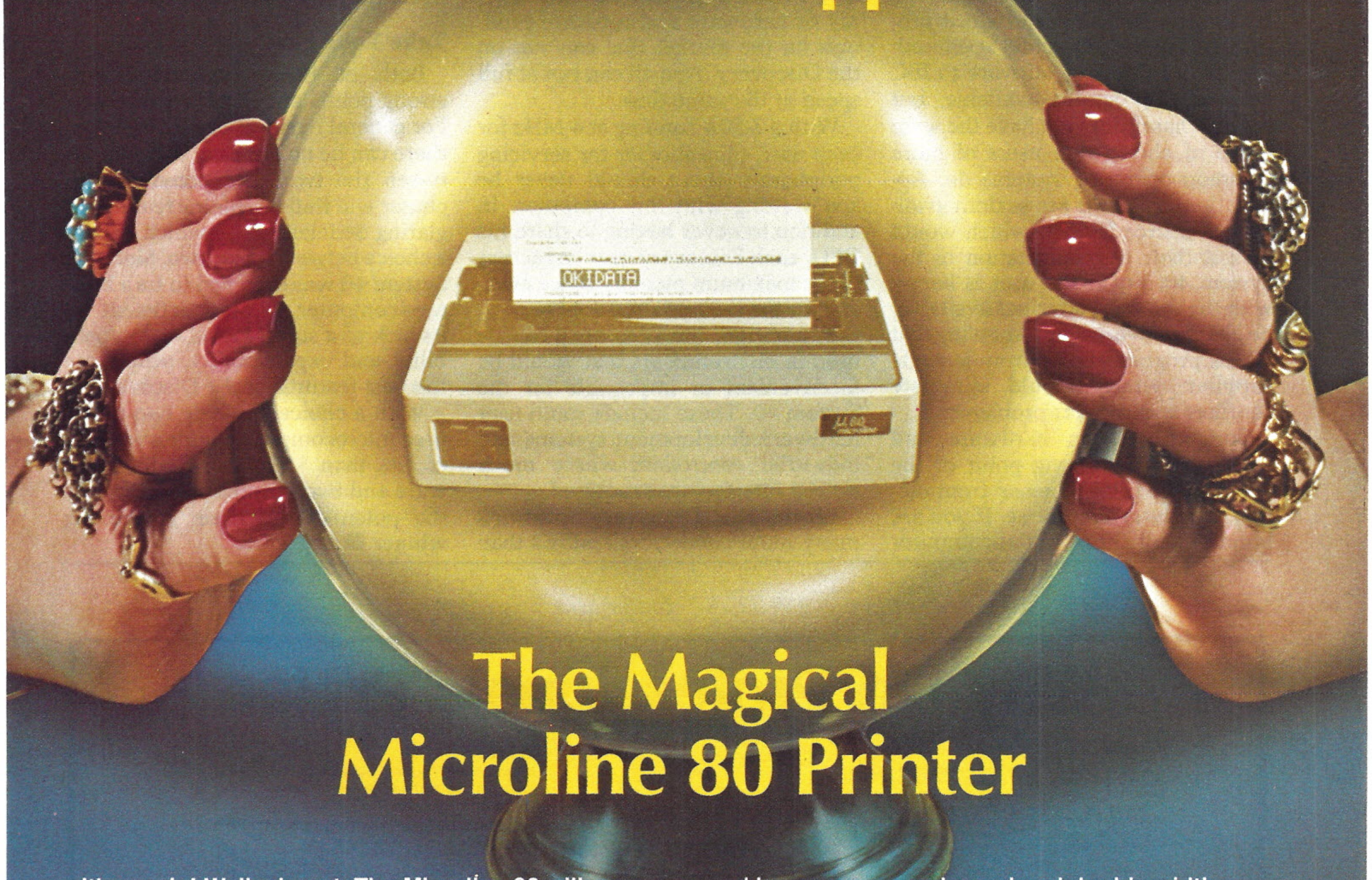
Software supported by the IBC Ensign includes MP/M, CP/M, Oasis and MVT-FAMOS. These operating systems represent the vast majority of Z-80-based software. With the Ensign, IBC introduces its new operating system: Superdos.

Supporting Business Basic and COBOL, Superdos combines with the Ensign to give minicomputer performance. Even when supporting 16 users under a heavy load, response for ISAM lookup on very large files will be under one second. Also, there is plenty of very good applications software running right now on expensive minicomputers supporting Business Basic and COBOL.

Up until now a computer dealer or end user had to rewrite all his software for a more expensive minicomputer if he wanted a larger system. With the Ensign all the software can be run unchanged with vastly improved performance.

The cost of the 64K byte Ensign with eight serial I/O ports and 2 megabytes of disk is $7000. The cost of expanded Ensign with 16 serial I/O ports, 768K memory, 70 megabyte disk and tape drive is approximately $33,000.

What has nine lives, three forms, multiple faces and a price tag that almost disappears?



The Magical Microline 80 Printer

It's magic! Well, almost. The Microline 80 will run all day at 80 cps with no duty cycle limitations. The head is warranted for 200,000,000 characters. That translates to over nine years on your TRS-80,™ APPLE® or other small computer.

Want to change forms? The magical Microline 80 is three printers disguised as one. There is a whisper-quiet rubber platen for cut sheets and roll paper, pins on nine inch centers for pin feed stock and optional snap-on tractors that adjust to suit all your other forms. The Microline 80 also saves paper by letting you tear off as close as one inch from the last print line.

Want to change your image? The magical Microline 80 really does tricks. It prints upper and lower case, condensed and double width characters *and* block graphics for charts, graphs and diagrams.

The Microline 80 is not a toy. With two motors, a rugged cast aluminum base and a head you never have to throw away, the Microline 80 is built to handle the most demanding business applications.

Which brings us to the biggest magic of all, the price tag, the one that almost disappears. If we're not the lowest, we are so close that it doesn't matter. There are stocking Microline distributors throughout the country. Call or write today for the name of the one near you and the price of the Magical Microline 80.

OKIDATA

365

Okidata Corporation, 111 Gaither Drive, Mount Laurel, New Jersey 08054 609-235-2600

Okidata is a subsidiary of Oki Electric Industry Company, Ltd.

will have to be slowed down at some time or other. On the brighter side, slow responses will be noticed most often at "logical" times, as when requesting a new program or more data from the disk.

The least desirable feature of the Integrated Business Computers system is its lack of compatibility with existing application software, such as accounting packages and word processors like WordStar and Spellbinder. The suppliers of software to IBC have such programs available, but the purchaser does not have the wide range of choice that users of more common operating systems do. Application programs are seldom available in source format, which would permit their easy adaptation to the System 40. But if the user is happy with the software offerings available, he has the advantage of knowing that all his programs come from a single supplier, which should eliminate conflicts and support problems.

Compatibility in hardware and software is the strong point of the Discovery Multiprocessor from Action Computer Enterprise. If you are familiar with the microcomputer scene, you know what S-100 and CP/M compatibility can mean in terms of the software availability and special-purpose peripherals.

Of course, in an industry with so many suppliers there are some bad eggs, and "compatible" sometimes does not mean just plug it in and go. But with some care in product selection, there is a virtually unlimited array of hardware and software products on the market that can run on the Discovery. And all can run at full speed at the same time.

With a Z-80A running at 4 MHz for each user, plus another for servicing peripherals, there should never be any waiting with this computer. In addition to never having to share his CPU, each Discovery user has access to the maximum memory space available in the Z-80 microprocessor.

In addition, the Discovery can be used in environments that would be inconvenient or impossible for the System 40. These include multi-user hardware development systems and industrial controllers where much special interfacing is required.

While the Discovery contains many more connectors and ICs than the IBC computer, reliability is not necessarily a problem. Since each user has identical hardware, the failure of a single component, say one of the Z-80s, would not mean that the entire system is inoperable. Performance can be degraded in the ACE computer by the loss of a single component, whereas the IBC machine could become totally inoperable.

Take Your Pick

Both of these computers have strong points and some weaknesses. For general business data processing, there can be no clear-cut decision between the two. The potential purchaser will find no shining features or glaring deficiencies on which to base a choice between these two. Since the System 40 was designed to be a business computer, it is not "deficient" in its lack of a similar level of compatibility and expandability, which are the best features of the Discovery.

With a choice between two multiuser microcomputers with price tags of less than $10,000 including software and two terminals, it looks like the potential purchaser is the big winner. ■

Announcing "Hellfire Warrior", a fantastic new Dunjonquest™ computer game... that's really not for everybody: Beginners are likely to be gobbled up in the first room...and there are over 200 rooms on four levels

"Hellfire Warrior." Really not for everybody: newcomers to Dunjonquest should begin with something easier. Here the monsters are deadlier, the labyrinths more difficult, the levels far more challenging...

But for the experienced Dunjonquest game player there are more command options, more potions (131), more magical items (including—at last—magical armor), more special effects, more surprises. And an innkeeper, an armorer, apothecary and magic shops.

In part a sequel to The Temple of Apshai, up until now the greatest of all the Dunjonquest games, Hellfire Warrior can also be played completely on its own.

Now the character you've created, representing the highest level of role-playing to date, can explore the four new lower levels:

Level 5—"The Lower Reaches of Apshai." With the giant insects and other nasties that swarmed through the upper levels of Apshai. With rooms your hero can get into, but not out of.

Level 6—"The Labyrinth." The only exit is hidden within the Labyrinth. And man-eating monsters can thwart your hero.

Level 7—"The Vault of the Dead."...And of the undead—skeletons, ghouls, mummies, specters...invisible ghosts—lurking in the rooms, doors, secret passages, ready to reduce your hero to a pale shadow of himself. Permanently.

Level 8—"The plains of Hell." In an Underworld of lost souls and shades of dead, of dragons and fiery hounds, of bottomless pits and blasts of hellfire, our hero must rescue the beautiful warrior maiden lying in enchanted sleep within a wall of fire. And bring her past unbelievable dangers and monsters...even Death itself...to sun and air and life itself.

Hellfire Warrior. The most exciting game yet from Automated Simulations, the leading producer of computer fantasy games.

Guaranteed: If it's not the most exciting computer game you've played, return it within ten days for a complete refund.

Available on disk for the Apple II and Radio Shack's TRS80, or on cassette for the TRS80 and for the Commodore PET. **The cassette or disk: $39.95.** Boxed with a magnificent instruction manual. Some of the drawings in the manual are reproduced here (in greatly reduced scale).

Hellfire Warrior. From the people who design and make some of the very, very best computer games, games that thinkers play. Including the great role-playing Dunjonquest series—The Temple of Apshai, Datestones of Ryn, Morloc's Tower, and now Hellfire Warrior. And Invasion Orion and Starfleet Orion. And more.

Hellfire Warrior. Like every EPYX computer game from Automated Simulations, it has the unique EPYX triple no-questions-asked warranty.

Hellfire Warrior. Now available on disk or cassette for TRS80, the Apple and the PET. At better computer stores everywhere. If you can't find it at your local friendly dealer, insist he order it for you.

Or you may order directly from Automated Simulations. **If you want to use your MasterCard or VISA, use our toll-free phones:** In the United States: operator 861 **(800) 824-7888;** In California: operator 861 **(800) 852-7777;** In Hawaii and Alaska: operator 861 **(800) 824-7919.** We'll also send you our new catalogue, hot off the press.

Or write Department F6, P.O. Box 4247, 1988 Leghorn St., Mountain View, CA 94040.

55

Please specify if you wish the **Disk** for TRS80 (32K, TRSDOS) and Apple (48K with Applesoft in ROM or the **Cassette** for the TRS80 (16K, Level II) or the Commodore PET (32K, old or new ROMs). Only $39.95, disk or cassette.

EPYX COMPUTER GAMES THINKERS PLAY

TRIPLE WARRANTY

10 DAY FULL MONEY BACK GUARANTEE: If you don't like the game for any reason whatever, return it intact within 10 days of receipt for a complete refund. No questions asked.

30 DAY DEFECTIVE WARRANTY: Cassette or disk not functioning within 30 days of receipt; return it for exchange. No charge, of course.

FOREVER—LIMITED LIFETIME WARRANTY: No matter what happens to your cassette or disk; the dog chewed it...you left it out in the rain...whatever. No matter **when** it happens. Return the remains to us (with $5.00 to cover all handling and shipping) and we'll send you a brand new cassette or disk.

How well do you know your North Star BASIC codes? Here's a practical analysis of coding conventions.

North Star Quiz

By Rinaldo F. Prisco

Here is a short quiz for North Star BASIC users:

1. What is the starting address of the programs in RAM?
2. What do the first three bytes of a program in RAM signify?
3. What is the end-of-program flag?
4. What is the format for coding line numbers referenced within a program?
5. How is the end of a programming line indicated?
6. When a new line is added to a program, is it placed at the end of the program or is it placed in sequential order, moving the other lines if necessary?
7. If a program is saved on disk, what do the first three bytes in the file signify?
8. Is it possible to open a BASIC program file from another BASIC program which treats it as a data file?
9. State precisely how the line:
120 IF A>B THEN 220
will be coded in hex.
10. "!" and "PRINT" signify the same thing. How many additional bytes of program space are required for each replacement of "!" by "PRINT"?

If you can answer at least seven of the above questions (even using references) in less than 20 minutes, skip this article. On the other hand, if you

STRING	DECIMAL	HEX	STRING	DECIMAL	HEX
!	146	92	INCHAR$	186	BA
(	224	E0	INP	217	D9
*	226	E2	INPUT	134	86
+	227	E3	LEN	204	CC
-	229	E5	LET	128	80
/	231	E7	LINE	156	9C
<	244	F4	LIST	161	A1
<=	240	F0	LOAD	165	A5
<>	241	F1	LOG	221	DD
=	245	F5	MEMSET	162	A2
>	246	F6	NEXT	131	83
>=	239	EF	NOENDMARK	185	B9
ABS	219	DB	NOT	247	F7
AND	236	EC	NSAVE	169	A9
ASC	182	B6	ON	147	93
ATN	210	D2	OPEN	151	97
AUTO	164	A4	OR	237	ED
CALL	205	CD	OUT	148	94
CAT	175	AF	PRINT	130	82
CHAIN	155	9B	PSIZE	174	AE
CHR$	181	B5	READ	133	85
CLOSE	152	98	REM	143	8F
CONT	166	A6	REN	168	A8
COS	220	DC	RESTORE	142	8E
CREATE	158	9E	RETURN	138	8A
DATA	135	87	RND	206	CE
DEF	145	91	RUN	160	A0
DEL	173	AD	SAVE	170	AA
DESTROY	157	9D	SCR	163	A3
DIM	139	8B	SIN	203	CB
ELSE	180	B4	SQRT	196	C4
END	141	8D	STEP	176	B0
ERRSET	159	9F	STOP	140	8C
EXAM	218	DA	STR$	184	B8
EXIT	150	96	TAB	179	B3
EXP	222	DE	THEN	178	B2
FILL	149	95	TO	177	B1
FN	144	90	TYP	223	DF
FOR	129	81	VAL	183	B7
GOSUB	137	89	WRITE	153	99
GOTO	136	88	^	225	E1
IF	132	84			

Table 1. North Star strings and codes.

```
0 REM REPLACE "!" BY "PRINT"
1 A=23612:E=146:P=130:G=154:C=13
2 A=A+1:IF EXAM(A)=1 THEN END
3 A=A+2
4 A=A+1:B=EXAM(A)
5 IF B=C THEN 2
6 IF B=G THEN 3
7 IF B=E THEN FILL A,P:GOTO 4
```

Listing 1. Replaces each occurrence of "!" by "PRINT" in any appended program.

Dr. Rinaldo F. Prisco (RD #7, Box 80, Oswego, NY 13126) is an Associate Professor of Mathematics for the State University of New York, College at Oswego.

```
0000                  * REPLACE "!" BY "PRINT"
0000                  * IN A PROGRAM IN BASIC
0000                  *
0000 21 92 5B                LXI   H,5B92H
0003 06 01                   MVI   B,1           END PROG FLAG
0005 0E 92                   MVI   C,146         "!" CODE
0007 16 82                   MVI   D,130         "PRINT" CODE
0009 1E 9A                   MVI   E,154         "GOTO" FLAG
000B 23               ST     INX   H
000C 7E                      MOV   A,M
000D B8                      CMP   B
000E C8                      RZ
000F 23               ST2    INX   H             SKIP LINE #'S
0010 23                      INX   H
0011 23               ST3    INX   H
0012 7E                      MOV   A,M
0013 FE 0D                   CPI   13            END OF LINE?
0015 CA 0B 00                JZ    ST
0018 BB                      CMP   E             A "GOTO"?
0019 CA 0F 00                JZ    ST2
001C B9                      CMP   C             A "!"?
001D C2 11 00                JNZ   ST3
0020 72                      MOV   M,D           REPLACE BY "PRINT"
0021 C3 11 00                JMP   ST3

SYMBOL TABLE

ST     000B     ST2    000F     ST3    0011
```

Listing 2. Replaces each occurrence of "!" by "PRINT" in any loaded BASIC program.

are curious about the answers and about how the answers might benefit you, then read on.

Codes

To conserve space and increase the execution speed, certain key strings in North Star BASIC are hashed into single-byte codes. Table 1 presents the sorted strings with corresponding codes in both decimal and hex. Table 2 provides the same information, but sorted by codes. Reference to these codes will help in reading or modifying a BASIC program in RAM.

Some applications which a knowledge of the coding conventions allows include recovery of most of a program that has been prematurely scratched, program sizes down to the precise byte, appending programs from RAM, variable and statement maps, program compactification, program locks, self-modifying programs and almost instantaneous changes of variables and statements.

The list can go on and on. Because of space limitations only a few of these applications will be discussed below. They are fairly straightforward programming procedures which rely on a knowledge of the code tables and many of the answers to the above quiz.

Simple Replacement

The tables indicate that the decimal codes for "!" and "PRINT" are 146 and 130, respectively. Thus, the answer to question 10 is that no additional bytes are required. So if we want to make our programs more readable to users of other BASICs, we can use "PRINT" in place of "!" without any additional space overhead. In fact, we don't even have to have the additional overhead of entering five characters in place of one.

If any BASIC program is appended to that in Listing 1 and then run, the replacement will be done throughout the program (except where "!" appears within quotes). A much faster assembly-language program which does the same thing appears in Listing 2.

The design of such programs requires the answers to the first four quiz questions. You must know where to begin your search (question 1), how to ignore bytes which reference line numbers (questions 2 and 4) and when to stop (question 3). The program area in Release 4 begins at 23443. The first byte in a line contains the number of bytes in that line; the next two bytes contain the line number. Line numbers referenced with a line (such as for GOTO, THEN, EXIT, GOSUB) are always prefaced by byte 154. The end of a line is indicated by 13, which is the

DECIMAL	HEX	STRING	DECIMAL	HEX	STRING
128	80	LET	174	AE	PSIZE
129	81	FOR	175	AF	CAT
130	82	PRINT	176	B0	STEP
131	83	NEXT	177	B1	TO
132	84	IF	178	B2	THEN
133	85	READ	179	B3	TAB
134	86	INPUT	180	B4	ELSE
135	87	DATA	181	B5	CHR$
136	88	GOTO	182	B6	ASC
137	89	GOSUB	183	B7	VAL
138	8A	RETURN	184	B8	STR$
139	8B	DIM	185	B9	NOENDMARK
140	8C	STOP	186	BA	INCHAR$
141	8D	END	196	C4	SQRT
142	8E	RESTORE	203	CB	SIN
143	8F	REM	204	CC	LEN
144	90	FN	205	CD	CALL
145	91	DEF	206	CE	RND
146	92	!	210	D2	ATN
147	93	ON	217	D9	INP
148	94	OUT	218	DA	EXAM
149	95	FILL	219	DB	ABS
150	96	EXIT	220	DC	COS
151	97	OPEN	221	DD	LOG
152	98	CLOSE	222	DE	EXP
153	99	WRITE	223	DF	TYP
155	9B	CHAIN	224	E0	(
156	9C	LINE	225	E1	^
157	9D	DESTROY	226	E2	*
158	9E	CREATE	227	E3	+
159	9F	ERRSET	229	E5	-
160	A0	RUN	231	E7	/
161	A1	LIST	236	EC	AND
162	A2	MEMSET	237	ED	OR
163	A3	SCR	239	EF	>=
164	A4	AUTO	240	F0	<=
165	A5	LOAD	241	F1	<>
166	A6	CONT	244	F4	<
168	A8	REN	245	F5	=
169	A9	NSAVE	246	F6	>
170	AA	SAVE	247	F7	NOT
173	AD	DEL			

Table 2. North Star codes and strings.

ASCII for CR.

You could, of course, modify the programs in Listing 1 and 2 for any replacements whatsoever. Poor typists could type in single-letter codes of their own for key codes and later run a program to make the replacement. If the key codes are permuted, then a BASIC program becomes incomprehensible unless one can decode it, giving some degree of privacy.

Complex Analysis

The replacement program is short enough so that it can work directly on a target program RAM. More complex modification or analysis of programs may require a good deal of space. What do you do if you have a large modification or analysis program, a large target program and limited RAM? Here is where the answer to quiz question 8 is needed. Indeed, the answer is yes! So you can access the target program as a data file, modify it and, if necessary, rewrite it to another file.

You can also extract information about the target program. The technique is to use READs instead of EXAMs, and WRITEs instead of FILLs. The GOTO program in Listing 3 uses this technique to get a table of all referenced line numbers in a target program, indicating the type of reference used (GOTO, GOSUB or EXIT).

The key bit of information which made this program easy to design was that all such line number references specify the line number by the two bytes which follow byte 154. So all you have to do is keep track of the source line number being scanned S(K) and look for byte 154. If found, you then record the referenced line in T(K) and backtrack (decrease pointer P) to see which type of statement made the reference: THEN (178), ELSE (180), GOTO (136), EXIT (150) or GOSUB (137).

The first three of them are variations of GOTO and are indicated in the printout in the same way. While EXIT is also a variation of GOTO, it is treated as a special case. Line 320 of the GOTO program codes the three cases: 1 for GOTO, ELSE or THEN; 2 for GOSUB; 3 for EXIT. These codes are then appended to the source line S(K).

On completion of the program scan, the target line numbers T(K) are sorted and the program passes to a simple printing routine for the formatted table.

Table 3 is the result of running the GOTO program on the dummy test program in Listing 4. Note that line 3 of the table verifies that line 250 of the dummy program is the target of lines 300, 700 and 800 with GOTOs, line 120 with GOSUB and line 190 with EXIT.

Concluding Remarks

We have discussed the codes and coding procedures used for North Star BASIC. These coding conventions were applied to the design of a simple replacement program and a more complex program which tabulates program flow.

An indication of the almost unlimited number of possible applications was also discussed. Although good

```
100 REM           EXTENDED GOTO
110 REM
120 REM            AUG. 1, 1980
130 REM
140 REM          RINALDO F. PRISCO
150 REM         MATHEMATICS DEPT.
160 REM      SUNY, COLLEGE AT OSWEGO
170 REM         OSWEGO, NY 13126
180 REM
190 DIM S(200),T(200)
200 K=1:G=154: REM - Line #'s prefaced by 154
210 INPUT "ENTER NAME OF BASIC PROGRAM: ",B$
220 OPEN #0%2,B$
230 READ #0,&L:P=P+1:IF L=1 THEN 340
240 READ #0,&D1,&D2:N=D1+256*D2:REM LINE #
250 P=P+2
260 READ #0,&C:P=P+1:IF C=13 THEN 230
270 IF C<>G THEN 260
280 READ #0,&D1,&D2:P=P+2:M=D1+256*D2
290 F=P-2

300 F=F-1:READ #0%F,&C
310 IFNOT(C=178ORC=136ORC=137ORC=150ORC=180)THEN300
320 C=1+(C=137)+2*(C=150)
330 T(K)=M:S(K)=N+C/10:K=K+1:READ #0%P,&C:GOTO 260
340 REM SORT
350 K=K-1:M=K
360 M=INT(M/2):IF M=0 THEN 450
370 J=1:N=K-M
380 I=J
390 L=I+M

400 IF T(I)<T(L) THEN 440
405 IF T(I)=T(L) THEN IF S(I)<=S(L) THEN 440
410 S=S(I):S(I)=S(L):S(L)=S
420 T=T(I):T(I)=T(L):T(L)=T
430 I=I-M:IF I>0 THEN 390
440 J=J+1:IF J>N THEN 360 ELSE 380
450 REM PRINT ROUTINE
460 INPUT "HARD COPY? ",Y$
470 IF Y$(1,1)="Y" THEN FILL 51207,2
480 !B$,"'s GOTOs":!:!
490 K=1
500 X=1:T=T(K):!
510 IF T=0 THEN 590
520 !%8I,T," <==",
530 !S(K),"  ",
540 K=K+1:X=X+1
550 IF T<>T(K) THEN 500
560 IF X<8 THEN 580
570 !:!TAB(15),:X=0
580 !S(K),"  ",:GOTO 540
590 FILL 51207,0:END
```

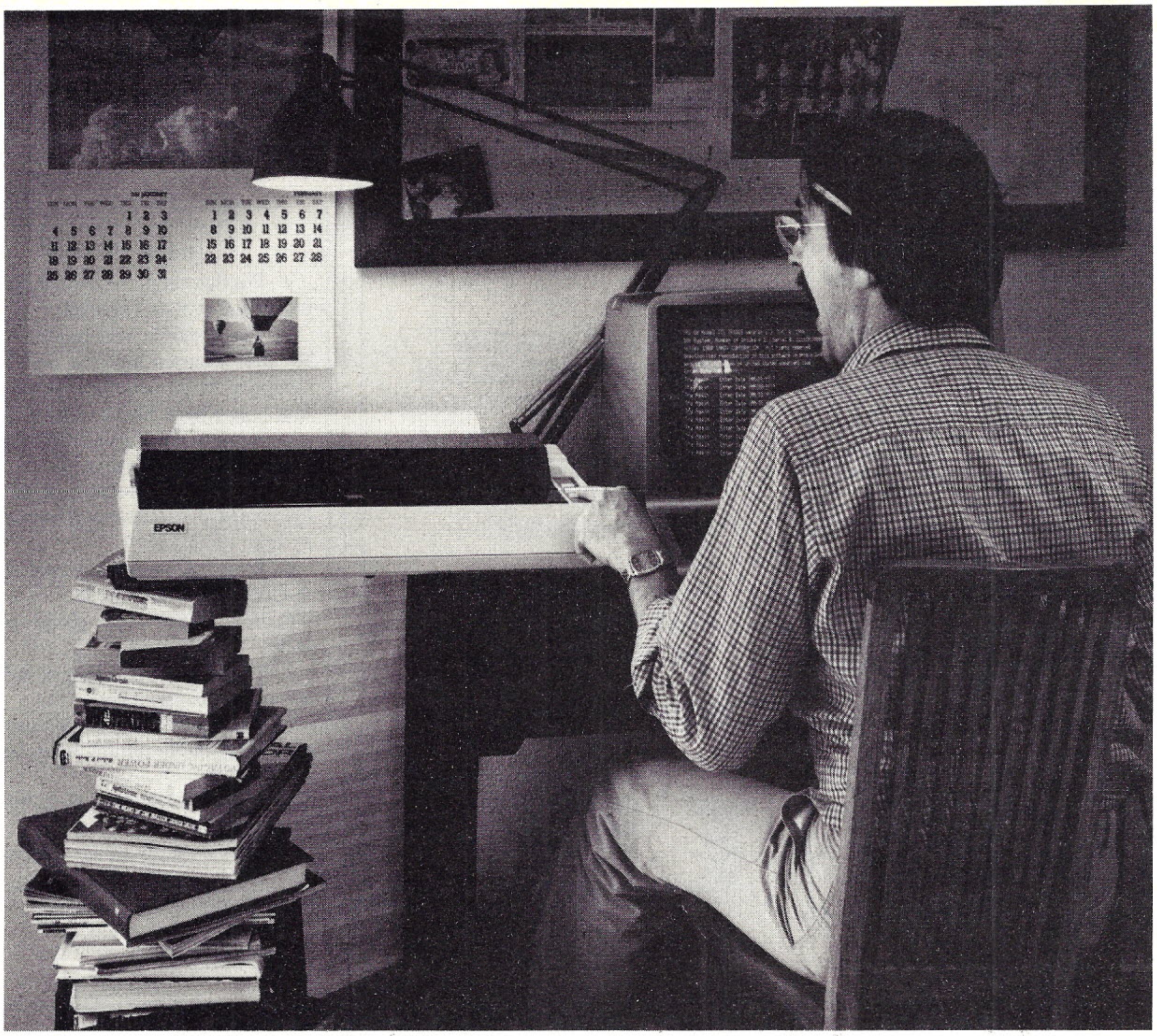
Listing 3. Provides an extensive GOTO map of any BASIC program on disk.

180	<==	190.1				
220	<==	200.1	900.1			
250	<==	120.2	190.3	300.1	700.1	800.1
530	<==	120.1	700.2			
600	<==	300.1	500.3	900.1		

Table 3. The GOTO table for the dummy test program in Listing 4.

```
100 REM DUMMY TEST PROGRAM
120 GOSUB 250:GOTO 530
190 GOTO 180: EXIT 250
200 GOTO 220
300 GOTO 600:GOTO 250
500 EXIT 600
700 GOTO 250:GOSUB 530
800 GOTO 250
900 IF A>B THEN 220 ELSE 600
```

Listing 4. A test program for Listing 3.



The MX-100. Not just better. Bigger.

Epson.

Our MX-80 was a pretty tough act to follow. I mean, how do you top the best-selling printer in the world?

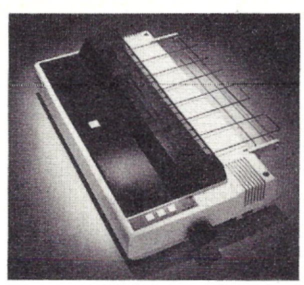
Frankly, it wasn't easy. But the results of all our sleepless nights will knock your socks off.

The MX-100 is a printer that must be seen to be believed. For starters, we built in unmatched correspondence quality printing, and an ultra-high resolution bit image graphics capability. Then we added the ability to print up to 233 columns of information on 15″ wide paper to give you the most incredible spread sheets you're ever likely to see. Finally, we topped it all off with *both* a satin-smooth friction feed platen *and* fully adjustable, removable tractors. And the list of standard features goes on and on and on.

Needless to say, the specs on this machine — and especially at under $1000 — are practically unbelievable. But there's something about the MX-100 that goes far beyond just the specs; something about the way it all comes together, the attention to detail, the fit, the feel. Mere words fail us. But when you see an MX-100, you'll know what we mean.

All in all, the MX-100 is the most remarkable printer we've ever built. Which creates rather a large problem for those of us at Epson.

How are we going to top this?



Your next printer.

EPSON

EPSON AMERICA, INC.

116

23844 Hawthorne Boulevard • Torrance, CA 90505 • (213) 378-2220

See the whole incredible Epson MX Series of printers at your Authorized Epson Dealer.

commercial software for the modification and analysis of North Star BASIC programs is available, no software can anticipate all possible needs of its users. There is just no substitute for know-how. ■

Quiz Answers

1. For Release 4, the first program byte is addressed at 23443 (5B93H).
2. The first byte is the number of bytes in the first program line; the next two bytes comprise its line number (in the 8080 format of low byte, high byte).
3. Program endings are indicated by the ASCII code SOH (byte 01).
4. All such references consist of two bytes as in (2) above; these bytes are flagged by byte 154 (9AH).
5. Program line endings are indicated by the ASCII code for CR, 13.
6. It is placed in its proper sequential order.
7. The same thing as they do in RAM; see the answer to (2) above.
8. Yes, use the North Star convention of specifying the file type (2 in this case) when OPENing it.
9. 10 78 00 20 84 20 41 F6 42 20 B2 20 9A C8 00 0D
10. None; they both use one byte.

The first personal computer for under $200.

The Sinclair ZX80. A complete computer— only $199.95 *plus $5.00 shipping.*

Now, for just $199.95, you can get a complete, powerful, full-function computer, matching or surpassing other personal computers costing several times more.

It's the Sinclair ZX80. The computer that "Personal Computer World" gave 5 stars for 'excellent value.'

The ZX80 cuts away computer jargon and mystique. It takes you straight into BASIC, the most common, easy-to-use computer language.

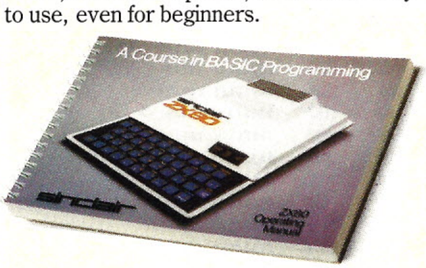
You simply take it out of the box, connect it to your TV, and turn it on. And if you want, you can use an ordinary cassette recorder to store programs. With the manual in your hand, you'll be running programs in an hour. Within a week, you'll be writing complex programs with confidence.

All for under $200.

Sophisticated design makes the ZX80 easy to learn, easy to use.

We've packed the conventional computer onto fewer, more powerful LSI chips—including the Z80A microprocessor, the faster version of the famous Z80. This makes the ZX80 the world's first truly portable computer (6½" x 8½" x 1½" and a mere 12 oz.). The ZX80 also features a touch sensitive, wipe-clean keyboard and a 32-character by 24-line display.

Yet, with all this power, the ZX80 is easy to use, even for beginners.



Your course in computing.

The ZX80 comes complete with its own 128-page guide to computing. The manual is perfect for both novice and expert. For every chapter of theory, there's a chapter of practice. So you learn by doing—not just by reading. It makes learning easy, exciting and enjoyable.

You'll also receive a catalog packed with items that can make your ZX80 even more useful. Including 27 program cassettes, from games and home budgeting for just $6.95, to Sinclair's unique Computer Learning Lab (a workbook, six cassettes with 100 lessons, and two cassettes for storing programs).

ZX80's advanced design features.

Sinclair's 4K integer BASIC has performance features you'd expect only on much larger and more expensive computers.

- Unique 'one touch' entry. Key words (RUN, PRINT, LIST, etc.) have their own single-key entry to reduce typing and save memory space.
- Automatic error detection. A cursor identifies errors immediately to prevent entering programs with faults.
- Powerful text editing facilities.
- Also programmable in machine code.
- Excellent string handling capability—up to 26 string variables of any length.
- Graphics, with 22 standard symbols.
- Built-in random number generator for games and simulations.

Sinclair's BASIC places no arbitrary restrictions on you—with many other flexible features, such as variable names of any length.

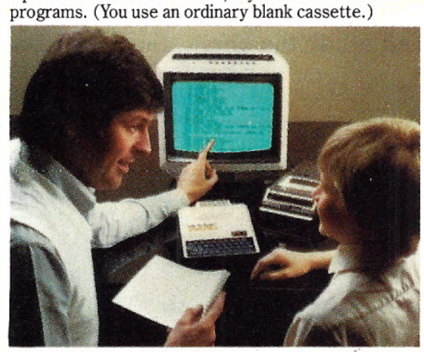
And the computer that can do so much for you now will do even more in the future. Options will include expansion of 1K user memory to 16K, a plug-in 8K floating-point BASIC chip, applications software, and other peripherals.

Order your ZX80 now!

The ZX80 is available only by mail from Sinclair, a leading manufacturer of consumer electronics worldwide.

To order by mail, use the coupon below. But for fastest delivery, order by phone and charge to your Master Charge or VISA. The ZX80 is backed by a 10-day money-back guarantee, and a 90-day limited warranty which can be extended by 12 months under Sinclair's extended service program for $25.00.

Price includes TV and cassette connectors, AC adaptor, and 128-page manual.
All you need to use your ZX80 is a standard TV (color or black and white). The ZX80 comes complete with connectors that easily hook up to the antenna terminals of your TV. Also included is a connector for a portable cassette recorder, if you choose to store programs. (You use an ordinary blank cassette.)



The ZX80 is a family learning aid. Children 10 and above will quickly understand the principles of computing—and have fun learning.

To order call toll free: 800-543-3000.
In Ohio call: 800-582-1364.
Ask for operator #508.
Phones open 24 hours a day, 7 days a week.

sinclair

Sinclair Research Ltd., One Sinclair Plaza, Nashua, NH 03061.

To: **Sinclair Research Ltd., One Sinclair Plaza, Nashua, NH 03061.**
Please send me ______ ZX80 personal computer(s) at $199.95 each (US dollars), plus $5 shipping. (Your ZX80 may be tax deductible.)
Send me ______ Computer Learning Lab(s) at $49.95 each.
Register me for ______ extended service program(s) at $25.00 each.
I enclose a check/money order payable to Sinclair Research Ltd. for $______.

Name ______
Address ______
City ______ State ______ Zip ______
Occupation ______ Age ______
Intended use of ZX80 ______
Have you ever used a computer? ☐ Yes ☐ No Do you own another personal computer? ☐ Yes ☐ No
0-6-KM

Despite inflation, your retirement can be financially comfortable, with proper planning and this Apple II application.

Plan Your Retirement On Easy Street

By G. R. Brieger

Death and taxes are two inescapable facts of life. We are faced today with a third—inflation. Thus, financial planning for retirement—though difficult and involved—is more important than ever.

VisiCalc software can help. It is ideal for solving "what if" problems. And while the application described here is for an Apple II+, VisiCalc is also available for the TRS-80, Commodore, Hewlett-Packard, Atari and PET microcomputers.

What Is VisiCalc?

VisiCalc is unique. Its manufacturer says that it is merely an electronic calculation sheet, but VisiCalc does a lot more. It could be called a limited enclosed computer language.

The program comprises 254 horizontal rows and 63 vertical columns for calculations. It provides over 16,000 locations for values, formulas

Address correspondence to G. R. Brieger, 2716 174th NE, Redmond, WA 98052.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	YEAR	SS HUSB.	SS WIFE	S.S.H+W	SPEND/	I.R.A.	WITHDRAW	LIQU.ASS.	TAXABLE	TAXES/	ASS. BEG.	% OF 1980	VALUE IN	GRAPHIC DISPLAY	
		% INCREASE		/MONTH	MONTH %	AT %	FROM IRA	AT %	INCOME	YEAR	OF YEAR	DOLLARS	1980 $'S	OF ASSETS FROM	
					6	11.00		11.00						COLUMN "K"	
6	1980	6.00	6.00		1500	100000		100000	11000	530	200000	100	1500	******** ******	
7	1981				1590	106000	5000	92470	15172	1200	198470	100	1590	******** *****	
8	1982				1685	112660	5000	87362	14610	1050	200022	100	1685	******** ******	
9	1983	560		560	1787	118053	7000	80697	15877	1200	198749	100	1787	******** *****	
10	1984	594		594	1894	122038	9000	80655	17872	1600	202693	100	1894	******** ******	
11	1985	629	315	944	2007	123463	12000	81326	20946	2200	204788	100	2007	******** ******	
12	1986	667	333	1000	2128	123043	14000	87309	23604	2900	210353	100	2128	******** ******	
13	1987	707	353	1060	2255	121578	15000	94485	25393	3400	216064	100	2255	******** *******	
14	1988	749	375	1124	2391	117952	17000	102139	28235	4200	220091	100	2391	******** *******	
15	1989	794	397	1192	2534	110927	20000	110975	32207	5300	221901	100	2534	******** *******	
16	1990	842	421	1263	2686	103129	20000	121770	33395	5600	224898	100	2686	******** ********	
17	1991	893	446	1339	3271	101082	13390	121770	26785	3600	222852	115	2847	******** *******	
18	1992	946	473	1419	3351	98811	13390	121770	26785	3600	220581	111	3018	******** *******	
19	1993	1003	501	1504	3436	96290	13390	121770	26785	3600	218060	107	3199	******** *******	
20	1994	1063	532	1595	3527	93492	13390	121770	26785	3600	215262	104	3391	******** *******	
21	1995	1127	563	1690	3622	90385	13390	121770	26785	3600	212155	101	3595	******** *******	
22	1996	1194	597	1792	3724	86938	13390	121770	26785	3600	208707	98	3811	******** ******	
23	1997	1266	633	1899	3831	83110	13390	121770	26785	3600	204880	95	4039	******** ******	
24	1998	1342	671	2013	3945	78862	13390	121770	26785	3600	200632	92	4282	******** ******	
25	1999	1423	711	2134	4066	74147	13390	121770	26785	3600	195917	90	4538	******** *****	
26	2000	1508	754	2262	4194	68913	13390	121770	26785	3600	190683	87	4811	******** *****	
27	2001	1598	799	2398	4330	63103	13390	121770	26785	3600	184873	85	5099	******** ****	
28	2002	1694	847	2542	4474	56654	13390	121770	26785	3600	178424	83	5405	******** ****	
29	2003	1796	898	2694	4626	49495	13390	121770	26785	3600	171265	81	5730	******** ***	
30	2004	1904	952	2856	4788	41549	13390	121770	26785	3600	163319	79	6073	******** ***	
31	2005	2018	1009	3027	4959	32730	13390	121770	26785	3600	154500	77	6438	******** **	
32	2006	2139	1070	3209	5141	22940	13390	121770	26785	3600	144709	75	6824	******** *	
33	2007	2267	1134	3401	5333	12073	13390	121770	26785	3600	133843	74	7234	********	
34	2008	2403	1202	3605	5537	10	13390	121770	26785	3600	121780	72	7668	********	
35	2009	2548	1274	3821	4867	6	5	121770	13400	850	121776	60	8128	********	
36	2010	2701	1350	4051	5096		0	121770	13395	850	121770	59	8615	********	
37								TOTAL INC.	TAX	95680					

Table 1. Sample retirement table. Tables 2–4 show results using different variables.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N O
	YEAR	SS HUSB.	SS WIFE	S.S.H+W	SPEND/	I.R.A.	WITHDRAW	LIQU.ASS.	TAXABLE	TAXES/	ASS. BEG.	% OF 1980	VALUE IN	GRAPHIC DISPLAY
		% INCREASE		/MONTH	MONTH %	AT %	FROM IRA	AT %	INCOME	YEAR	OF YEAR	DOLLARS	1980 $'S	OF ASSETS FROM
					5	11.00		11.00						COLUMN "K"
6	1980	5.00	5.00		1500	100000		100000	11000	530	200000	100	1500	******** *****
7	1981				1575	106000	5000	92470	15172	1200	198470	100	1575	******** *****
8	1982				1654	112660	5000	87542	14630	1050	200202	100	1654	******** *****
9	1983	560		560	1736	118053	7000	81276	15940	1200	199329	100	1736	******** *****
10	1984	588		588	1823	122038	9000	81899	18009	1800	203938	100	1823	******** *****
11	1985	617	309	926	1914	123463	12000	83285	21161	2400	206748	100	1914	******** *****
12	1986	648	324	972	2010	123043	14000	90187	23921	2900	213230	100	2010	******** ******
13	1987	681	340	1021	2111	121578	15000	98754	25863	3400	220333	100	2111	******** ******
14	1988	715	357	1072	2216	117952	17000	108142	28896	4200	226094	100	2216	******** *******
15	1989	750	375	1126	2327	110927	20000	119108	33102	5600	230035	100	2327	******** *******
16	1990	788	394	1182	2443	103129	20000	132194	34541	6000	235323	100	2443	******** ********
17	1991	827	414	1241	3244	101083	13390	132194	27931	3900	233277	126	2566	******** *******
18	1992	869	434	1303	3306	98812	13390	132194	27931	3900	231006	123	2694	******** *******
19	1993	912	456	1368	3371	96291	13390	132194	27931	3900	228485	119	2828	******** *******
20	1994	958	479	1437	3439	93493	13390	132194	27931	3900	225687	116	2970	******** *******
21	1995	1006	503	1509	3511	90387	13390	132194	27931	3900	222582	113	3118	******** *******
22	1996	1056	528	1584	3587	86940	13390	132194	27931	3900	219134	110	3274	******** ******
23	1997	1109	554	1663	3666	83113	13390	132194	27931	3900	215308	107	3438	******** ******
24	1998	1164	582	1746	3749	78866	13390	132194	27931	3900	211060	104	3610	******** ******
25	1999	1222	611	1834	3836	74151	13390	132194	27931	3900	206345	101	3790	******** *****
26	2000	1284	642	1925	3928	68918	13390	132194	27931	3900	201112	99	3980	******** *****
27	2001	1348	674	2022	4024	63109	13390	132194	27931	3900	195303	96	4179	******** ****
28	2002	1415	708	2123	4125	56660	13390	132194	27931	3900	188855	94	4388	******** ****
29	2003	1486	743	2229	4231	49503	13390	132194	27931	3900	181698	92	4607	******** ***
30	2004	1560	780	2340	4343	41558	13390	132194	27931	3900	173753	90	4838	******** ***
31	2005	1638	819	2457	4460	32740	13390	132194	27931	3900	164934	88	5080	******** **
32	2006	1720	860	2580	4583	22951	13390	132194	27931	3900	155146	86	5334	******** *
33	2007	1806	903	2709	4712	12086	13390	132194	27931	3900	144280	84	5600	********
34	2008	1896	948	2845	4847	25	13390	132194	27931	3900	132220	82	5880	********
35	2009	1991	996	2987	4112	21	7	132194	14548	1050	132216	67	6174	********
36	2010	2091	1045	3136	4260		0	132194	14541	1050	132194	66	6483	********
37									TOTAL INC. TAX	102580				

Table 2.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N O
	YEAR	SS HUSB.	SS WIFE	S.S.H+W	SPEND/	I.R.A.	WITHDRAW	LIQU.ASS.	TAXABLE	TAXES/	ASS. BEG.	% OF 1980	VALUE IN	GRAPHIC DISPLAY
		% INCREASE		/MONTH	MONTH %	AT %	FROM IRA	AT %	INCOME	YEAR	OF YEAR	DOLLARS	1980 $'S	OF ASSETS FROM
					5	11.00		11.00						COLUMN "K"
6	1980	5.00	5.00		1600	100000		100000	11000	530	200000	100	1600	******** ******
7	1981				1680	106000	5000	91270	15040	1200	197270	100	1680	******** ******
8	1982				1764	112660	5000	84950	14344	1050	197610	100	1764	******** ******
9	1983	560		560	1852	118053	7000	77076	15478	1200	195129	100	1852	******** ******
10	1984	588		588	1945	122038	9000	75848	17343	1600	197887	100	1945	******** ******
11	1985	617	309	926	2042	123463	12000	75310	20284	2200	198772	100	2042	******** ******
12	1986	648	324	972	2144	123043	14000	80002	22800	2700	203046	100	2144	******** *******
13	1987	681	340	1021	2251	121578	15000	86042	24465	3000	207620	100	2251	******** *******
14	1988	715	357	1072	2364	117952	17000	92742	27202	3900	210694	100	2364	******** *******
15	1989	750	375	1126	2482	110927	20000	100542	31060	5000	211468	100	2482	******** *******
16	1990	788	394	1182	2606	103129	20000	110324	32136	5300	213452	100	2606	******** ********
17	1991	827	414	1241	3085	101082	13390	110324	25526	3400	211406	113	2737	******** *******
18	1992	869	434	1303	3147	98811	13390	110324	25526	3400	209135	110	2873	******** *******
19	1993	912	456	1368	3212	96290	13390	110324	25526	3400	206614	106	3017	******** *******
20	1994	958	479	1437	3281	93491	13390	110324	25526	3400	203815	104	3168	******** *******
21	1995	1006	503	1509	3352	90385	13390	110324	25526	3400	200709	101	3326	******** *******
22	1996	1056	528	1584	3428	86937	13390	110324	25526	3400	197261	98	3493	******** ******
23	1997	1109	554	1663	3507	83109	13390	110324	25526	3400	193433	96	3667	******** ******
24	1998	1164	582	1746	3590	78861	13390	110324	25526	3400	189185	93	3851	******** ******
25	1999	1222	611	1834	3677	74145	13390	110324	25526	3400	184469	91	4043	******** *****
26	2000	1284	642	1925	3769	68911	13390	110324	25526	3400	179235	89	4245	******** *****
27	2001	1348	674	2022	3865	63101	13390	110324	25526	3400	173425	87	4458	******** ****
28	2002	1415	708	2123	3966	56651	13390	110324	25526	3400	166975	85	4680	******** ****
29	2003	1486	743	2229	4073	49493	13390	110324	25526	3400	159817	83	4914	******** ***
30	2004	1560	780	2340	4184	41546	13390	110324	25526	3400	151870	81	5160	******** ***
31	2005	1638	819	2457	4301	32726	13390	110324	25526	3400	143050	79	5418	******** **
32	2006	1720	860	2580	4424	22936	13390	110324	25526	3400	133260	78	5689	******** *
33	2007	1806	903	2709	4553	12068	13390	110324	25526	3400	122392	76	5974	********
34	2008	1896	948	2845	4688	5	13390	110324	25526	3400	110329	75	6272	********
35	2009	1991	996	2987	3940	1	5	110324	12141	700	110325	60	6586	********
36	2010	2091	1045	3136	4089		0	110324	12136	700	110324	59	6915	********
37									TOTAL INC. TAX	90280				

Table 3.

and labels. This both saves time and avoids errors when setting up tables.

Once you establish the format, VisiCalc fills the columns automatically. The math is done in seconds right before your eyes, and the results can be output on your printer.

The accompanying sample tables show the information entered, including the constants used. VisiCalc recalculates the data instantly when any data, such as inflation rates or earnings, is changed. The beginning dollar amounts in the columns can be varied to suit individual investments or pensions.

The Tables Explained

The tables (1–6) assume a retired man born in 1920 with a wife two years younger. He plans to retire early, at age 60. He has about $100,000 in savings and investments, and $100,000 in an IRA or Keogh account which can be rolled over.

This rollover account is tax-sheltered until he reaches age 70 1/2 (1991). At that time, according to present Internal Revenue Service rules, he must liquidate the account in about 18 years. The money is fully taxable as withdrawn. Withdrawals can be made earlier, and possibly should be, to minimize income taxes.

Here is a detailed explanation of the tables, by column:

Column A shows the years from 1980 to 2010; on the far left are line numbers for easy location of data points.

Columns B, C and D contain social security payments. Changing constants result in recalculation of inflation/COLA (cost of living adjustment) increases or investment returns. The percentage increases used also show up automatically at the top of columns B, C and E, and F and H, respectively.

Column E represents the whole reason of this exercise: spendable dollars after taxes. Annual increases to keep up with inflation can be varied by different multipliers. The increase is a straight percentage until 1990. After that the amounts are determined by what is available from columns D, G and H. E6 is the starting amount of personal budget needs.

Columns F and H contain the amounts of assets, tax-sheltered and liquid, respectively, as mentioned above. By entering different amounts in locations F6 and H6, individual personal financial conditions will be displayed instead of the examples shown here.

Column G is for indicating amounts withdrawn from the tax-sheltered rollover account. While no withdrawals are required until 1991 by the IRS, minimizing income taxes indicates increasing payouts from this account as shown. Amounts from G6 to G16 are derived somewhat by trial and error to keep income taxes as low as possible and also to conserve sufficient liquid capital in column H. The quantity in location G17 has to be calculated from the dollar amount in F16. It is the amount needed to liquidate the tax-sheltered account starting in 1991.

Column I is the sum of column G plus earnings of column H.

Column J shows income taxes entered from special tables, which are shown together with the constants in Table 6. This table is strictly for the technical-minded. The Lookup Function of VisiCalc fills in the taxes. Total income taxes paid out from 1980 to 2010 are added into location J37, with the label indicating this.

Columns K and L are added to give

	A	B	C	D	E	F	G	H	I	J	K	L	M	N O
	YEAR	SS HUSB.	SS WIFE	S.S.H+W	SPEND/	I.R.A.	WITHDRAW	LIQU.ASS.	TAXABLE	TAXES/	ASS. BEG.	% OF 1980	VALUE IN	GRAPHIC DISPLAY
		% INCREASE		/MONTH	MONTH %	AT %	FROM IRA	AT %	INCOME	YEAR	OF YEAR	DOLLARS	1980 $'S	OF ASSETS FROM
					5.50	12.00		11.00						COLUMN "K"
6	1980	5.50	5.50		1600	100000		100000	11000	530	200000	100	1600	******** ******
7	1981				1688	107000	5000	91270	15040	1200	198270	100	1688	******** ******
8	1982				1781	114840	5000	84854	14334	1050	199694	100	1781	******** ******
9	1983	560		560	1879	121621	7000	76768	15444	1200	198388	100	1879	******** ******
10	1984	591		591	1982	127215	9000	75187	17271	1600	202402	100	1982	******** ******
11	1985	623	312	935	2091	130481	12000	74161	20158	2200	204642	100	2091	******** ******
12	1986	658	329	986	2206	132139	14000	78245	22607	2700	210383	100	2206	******** ******
13	1987	694	347	1041	2327	132996	15000	83514	24187	3000	216510	100	2327	******** *******
14	1988	732	366	1098	2455	131955	17000	89258	26818	3600	221213	100	2455	******** *******
15	1989	772	386	1158	2591	127790	20000	96185	30580	4700	223974	100	2591	******** *******
16	1990	815	407	1222	2733	123124	20000	104877	31536	5000	228001	100	2733	******** ********
17	1991	859	430	1289	3316	120916	16983	104877	28520	4200	225793	115	2883	******** *******
18	1992	907	453	1360	3387	118443	16983	104877	28520	4200	223320	111	3042	******** *******
19	1993	957	478	1435	3461	115673	16983	104877	28520	4200	220550	108	3209	******** *******
20	1994	1009	505	1514	3540	112570	16983	104877	28520	4200	217447	105	3386	******** *******
21	1995	1065	532	1597	3624	109095	16983	104877	28520	4200	213973	101	3572	******** *******
22	1996	1123	562	1685	3711	105204	16983	104877	28520	4200	210081	98	3768	******** ******
23	1997	1185	593	1778	3804	100845	16983	104877	28520	4200	205722	96	3976	******** ******
24	1998	1250	625	1875	3902	95963	16983	104877	28520	4200	200840	93	4194	******** ******
25	1999	1319	659	1978	4005	90496	16983	104877	28520	4200	195373	91	4425	******** *****
26	2000	1391	696	2087	4114	84372	16983	104877	28520	4200	189249	88	4668	******** *****
27	2001	1468	734	2202	4229	77513	16983	104877	28520	4200	182390	86	4925	******** *****
28	2002	1549	774	2323	4350	69832	16983	104877	28520	4200	174709	84	5196	******** ****
29	2003	1634	817	2451	4478	61228	16983	104877	28520	4200	166105	82	5482	******** ***
30	2004	1724	862	2586	4612	51592	16983	104877	28520	4200	156470	80	5783	******** ***
31	2005	1819	909	2728	4755	40800	16983	104877	28520	4200	145678	78	6101	******** **
32	2006	1919	959	2878	4905	28713	16983	104877	28520	4200	133590	76	6437	******** *
33	2007	2024	1012	3036	5063	15176	16983	104877	28520	4200	120053	75	6791	********
34	2008	2135	1068	3203	5230	13	16983	104877	28520	4200	104891	73	7165	********
35	2009	2253	1126	3379	4297	7	8	104877	11544	530	104884	57	7559	********
36	2010	2377	1188	3565	4483		0	104877	11536	530	104877	56	7974	********
37								TOTAL	INC. TAX	103440				

Table 4.

an overall picture of the retirement status in any year for which a future prediction has been made. Column K shows that total net worth stays almost even until at least 1990. After that it declines slowly (just as one gets older, requiring less). Column L holds percentage equivalents of 1980 income after taxes.

Columns N and O in Tables 1–4 show a simple visual display of assets at the beginning of each year.

Table 5 is an example of retirement with a pension, smaller tax-sheltered plan and personal savings. Column N contains the amount of the pension. Such a table might be used by corporate planners wanting to show employees what to expect financially during retirement years.■

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	YEAR	SS HUSB. % INCREASE	SS WIFE	S.S.H+W /MONTH	SPEND/ MONTH %	I.R.A. AT %	WITHDRAW FROM IRA	LIQU.ASS. AT %	TAXABLE INCOME	TAXES/ YEAR	ASS. BEG. OF YEAR	% OF 1980 DOLLARS	VALUE IN 1980 $'S	PENSION/ MONTH
					6	11.00		11.00						
6	1980	6.00	6.00		1100	30000		40000	11600	530	70000	100	1100	600
7	1981				1166	32050	1250	37870	12616	700	69920	100	1166	600
8	1982				1236	34076	1500	35794	12637	700	69869	100	1236	600
9	1983	560		560	1310	36074	1750	32899	12569	700	68973	100	1310	600
10	1984	594		594	1389	38042	2000	35767	13134	850	73809	100	1389	600
11	1985	629	315	944	1472	39727	2500	38510	13936	850	78236	100	1472	600
12	1986	667	333	1000	1560	41096	3000	45257	15178	1200	86354	100	1560	600
13	1987	707	353	1060	1654	41867	3750	52517	16727	1400	94384	100	1654	600
14	1988	749	375	1124	1753	41972	4500	60721	18379	1800	102694	100	1753	600
15	1989	794	397	1192	1858	41089	5500	69751	20373	2200	110841	100	1858	600
16	1990	842	421	1263	1970	38609	7000	79921	22991	2700	118531	100	1970	600
17	1991	893	446	1339	2972	37843	5013	91730	22303	2700	129573	142	2088	600
18	1992	946	473	1419	3053	36993	5013	91730	22303	2700	128723	138	2213	600
19	1993	1003	501	1504	3138	36049	5013	91730	22303	2700	127779	134	2346	600
20	1994	1063	532	1595	3228	35002	5013	91730	22303	2700	126732	130	2487	600
21	1995	1127	563	1690	3324	33839	5013	91730	22303	2700	125569	126	2636	600
22	1996	1194	597	1792	3425	32548	5013	91730	22303	2700	124278	123	2794	600
23	1997	1266	633	1899	3533	31115	5013	91730	22303	2700	122845	119	2962	600
24	1998	1342	671	2013	3647	29525	5013	91730	22303	2700	121255	116	3140	600
25	1999	1423	711	2134	3768	27760	5013	91730	22303	2700	119490	113	3328	600
26	2000	1508	754	2262	3896	25800	5013	91730	22303	2700	117531	110	3528	600
27	2001	1598	799	2398	4031	23626	5013	91730	22303	2700	115356	108	3740	600
28	2002	1694	847	2542	4175	21211	5013	91730	22303	2700	112941	105	3964	600
29	2003	1796	898	2694	4328	18532	5013	91730	22303	2700	110262	103	4202	600
30	2004	1904	952	2856	4489	15557	5013	91730	22303	2700	107287	101	4454	600
31	2005	2018	1009	3027	4661	12255	5013	91730	22303	2700	103985	99	4721	600
32	2006	2139	1070	3209	4842	8590	5013	91730	22303	2700	100320	97	5004	600
33	2007	2267	1134	3401	5035	4522	5013	91730	22303	2700	96252	95	5305	600
34	2008	2403	1202	3605	5239	7	5013	91730	22303	2700	91737	93	5623	600
35	2009	2548	1274	3821	5129	3	5	91730	17295	1600	91733	86	5960	600
36	2010	2701	1350	4051	5358		0	91730	17290	1600	91730	85	6318	600
37								TOTAL INC. TAX		65430				

Table 5. Sample retirement table with pension factor.

INCOME TAX TABLES

10000	370
11000	530
12000	700
13000	850
14000	1050
15000	1200
16000	1400
17000	1600
18000	1800
19000	2000
20000	2200
21000	2400
22000	2700
23000	2900
24000	3000
25000	3400
26000	3600
27000	3900
28000	4200
29000	4500
30000	4700
31000	5000
32000	5300
33000	5600
34000	6000
35000	6300
36000	6700
37000	7000
38000	7500
39000	7700
40000	8000

```
39   CONSTANTS
40 CONST.     1.00 FOR CALC
41 CONST. .0091667 =(B45-1)

43 CONST.     1.06 FOR INCREASING S.S. AND INCOME
44 CONST.     1.11 INCOME ON I.R.A.
45 CONST.     1.11 INCOME ON LIQ.ASS.


48   FORMULAS
49 B6 AND C6 = B43-1*100
50 B10 = B9*B43
51 C11 = B11/2
52 D 9 = B9+C9
53 D12 = B12+C12+200
54 E 8 = E10*B43
55 F 5 = B44-1*100
56 H 5 = B45-1*100
57 G18 = G17*B40
58 I 6 = H6*H5/12+G6
59 K 6 = F6+H6
60 M 8 = M7*B43
61 L 6 = E6*100/L6
62 H 7 = (H6*B45)-(E6*12)-J6+(D6*12)+G6
63 E17 = D17+(I17/12)-(J17/12)
64 H17 = H16*B40
```

Table 6. Income tax table with constants.

✓See List of Advertisers on page 210

DIABLO 1620 $1895

- Letter quality daisywheel printer
- Upper and lower case
- Forms tractor, print wheel, ribbon included
- ASCII serial interface with RS-232 cable
- Bidirectional printing
- Completely refurbished by national terminal distributor with 30 day warranty

PRINTERS

NEC 5510	$2495
NEC 5530	2495
Diablo 630	2425
Centronics 730	675
Centronics 737	799

VIDEO MONITORS

9" Sanyo B&W	$169
12" Sanyo B&W	289
12" Sanyo w/green screen	299
13" Sanyo Color	495
9" Panasonic B&W	169
9" RCA B&W	169

DISKETTES

Verbatim			3M		
5"SS,DD	10/2.40	50/2.35	5"SS,SD	10/2.80	50/2.70
5"DS,DD	10/3.95	50/3.90	8"SS,SD	10/2.90	50/2.80
5"SS,QD	10/2.90	50/2.85	8"SS,DD	10/3.50	50/3.40
8" SS, SD	10/3.05	50/3.00	8" DS, SD	10/4.50	50/4.40
8" SS, DD	10/3.35	50/3.30	8" DS, DD	10/4.50	50/4.40
8" DS, DD	10/4.15	50/4.10	Disk Flip 'N File Case 5" 22.95, 8" 27.95		

Diskette Storage Pages 10/4.25 — Disk Library Case 5" 2.05, 8" 2.80

3M HEAD CLEANING KIT

- Eliminate downtime
- Eliminate service calls
- Increase life of read-write heads
- Cleans in 30 seconds
- Removes dust, dirt, magnetic oxides

5¼" or 8" **$19.50**

PAPER & LABELS

Greenbar 14⅞x11	
one part 3000 sheets	39.95
two part 1300 sheets	45.95
White 9½x11	
one part 3500 sheets	29.95
two part 1400 sheets	33.95
Labels 3½x15/16	
one across box of 5000	17.30
two across box of 10,000	34.60

PRINT WHEELS

Mfr.	ea.	6+
Diablo	8.25	7.50
Qume	8.25	7.50
NEC	17.00	15.50

RIBBONS

Mfr.	ea.	Doz.
Diablo	6.50	72.00
Qume	4.00	43.50
NEC	7.00	71.50

Call for quantity prices—Minimum $1.75 shipping and handling. ✓63

DSI
DAVIS SYSTEMS INC.
2184 Meadowcliff Drive N.E., Atlanta, GA. 30345
(404) 634-2300

---

computer case company

comp case

- AP104

• AP101	Apple II with Single Disk Drive	$109
• AP102	Apple II with Double Disk Drives	119
• AP103	Apple II, 9 inch Monitor & Double Drives	129
• AP104	Apple ///, two additional Drives & Silentype	139
• AP105	12 inch monitor plus accessories	99
• RS201	TRS-80 Model I, Expansion Unit & Drives	109
• RS202	TRS-80 Monitor or TV set	84
• RS204	TRS-80 Model III	129
• RS205	Radio Shack Color Computer	89
• P401	Paper Tiger 440/445/460	99
• P402	Centronics 730/737 - Line Printer II/IV	89
• P403	Epson MX70 or MX80	89
• CC90	Matching Attaché Case	75

computer case company ✓320

5650 INDIAN MOUND CT. COLUMBUS, OHIO 43213 (614) 868-9464

WHY USE OUR INTEGRATED ACCOUNTING SYSTEM (IAS)?

IT'S EASY TO USE. Prompts guide the user through each entry and then checks the entry to make sure only proper data are saved on the disk. It won't let you post an out-of-balance entry. IAS includes a complete users manual that tells you what each program does and how to use it. We even include a practice disk so you can begin making entries within minutes after receiving our IAS.

IT'S FAST. Our proprietary Skip Sequential™ file structure lets you save data on the disk at maximum speed without wasting disk space. We've also minimized keystrokes to make more efficient use of your operator's time.

IT GIVES YOU THE INFORMATION YOU NEED. The complete IAS package has over 30 reports, listings and file printouts so you can monitor the firm's fiscal position at all times, including instantaneous account balances. Our efficient file structure also means there's no monthly "erasing" of data files. You'll appreciate this if you're ever required to reconstruct an audit trail for some past month.

WE SUPPORT OUR SOFTWARE. Since IAS is not a translation of someone else's work, but was written by us from scratch, we know it cold. If you ever have a problem, we can answer it for you. We also give you full credit on any future packages that might replace the one you purchased. Our record over the past three years has been a good one, and we'll do everything we can to keep you a satisfied customer.

PRICE. The General Ledger costs $125.00; the complete IAS package (GL, AP, AR and PR) is $350.00. The user's manual is $20.00 (refunded with purchase). Since IAS is designed for the North Star DOS and Basic and will run in as little as 32K of memory, you don't need to buy more memory, a different operating system and Basic. This savings alone could pay for the software—and then some. You can pay a lot more, but shouldn't you check us out before you do?

master charge THE INTERBANK CARD

ECOSOFT ✓82
P.O. Box 68602
Indianapolis, IN 46268
(317) 283-8883
(Phone Orders Only)

DEALERS... SELL

kilobaud MICROCOMPUTING

Selling Kilobaud MICROCOMPUTING, the most complete journal of microcomputing, brings the computer enthusiast through your door. Once he's in your store, you can sell him anything.

For information on selling Kilobaud MICROCOMPUTING, call 603-924-7296 and speak with Ginnie Boudrieau, our bulk sales manager, or write to her at Kilobaud Microcomputing, 80 Pine Street, Peterborough, NH 03458.

Our dealers are telling us that Kilobaud MICROCOMPUTING is the hottest-selling computer magazine on the newsstand, so call today and join the ranks of dealers who make money with KM.

kilobaud MICROCOMPUTING
80 PINE STREET PETERBOROUGH NH 03458

Double and quadruple your cassette storage speed with this circuit for the OSI Superboard and C1P.

OSI Baud Mod

By Ronald W. Carr

There are a number of reasons why I built this circuit mod for my Superboard. I had spent several hours debugging an assembly-language program that was too long to assemble in one piece, and most of those hours were spent loading and saving pieces of source and object code at 300 baud. I knew there were four areas on the board marked proto—and they had to be there for some reason. I had an idea for a modification to speed up loading and saving, and I had to see whether it would work. Construction began the next day.

I was hesitant about soldering on my new Superboard, but I'm glad I proceeded. The modification is not difficult if you have some experience working on circuit boards. Four ICs are used, but some gates aren't, so not too much work is involved. Parts, including sockets for the ICs, can be bought at Radio Shack for under $10. Construction will probably take about four hours if you are as careful as I was when building. Then you will be saving BASIC programs at 600 baud—twice as fast.

Design

The basic design principle of the circuit is simple—in a word, redundant. The existing circuit that clocks the ACIA serial port is duplicated, except for the addition of a multiplexer that allows the selection of different input clocks to the ACIA.

There are good reasons for doing things this way. The original circuit supplies a clock to the Kansas City transmitter as well as to the ACIA. The clock to the transmitter must remain at 4800 Hz, or the tones that are recorded will be too high a frequency for both the recorder and the Kansas City receiver circuit.

Duplicating the circuit allows for separate clocking sources for the ACIA and the transmitter circuit. These separate clocks remain synchronous (i.e., they change states simultaneously) because the circuit *is* duplicated and because the existing divider chain is synchronous. For proper operation the clocks must be synchronous.

The other reason I chose to duplicate the existing circuit is that foil cuts are required only at W5, a hardwire jumper section included on the board so that you can easily change the clock to the ACIA. Thus, the original circuitry is kept intact.

Let's look at Fig. 1. The signals C2, C3 and C4 are obtained from the existing divider chain. I'll tell you good places to find these later. U44 is a data selector or multiplexer which, depending upon the position of SW1, selects C2, C3 or C4 for a clock to U26. U26 and U27 form a divide-by-13 counter. State 12 is decoded by U27, and, when the next clock pulse arrives, all zeros are loaded using the parallel inputs. U28 divides the output of U26 by two, and makes the waveform symmetrical. The output of U28 is used as the new clock for the 6850 ACIA.

Construction

I suggest that care be taken during construction so that any problems during checkout will be simple to solve. I labeled pin 1 of each IC on the bottom of the board with a felt-tip pen, and also labeled the +5 and ground foil runs. You should verify these with an ohmmeter before you start. Remove the board from its case, and label any wires that need to be disconnected to aid in reassembly.

74LS161—Install a 16-pin socket at the proto area designated U26 (refer to the user's manual for the location of U26). Pin 1 is located toward the keyboard. Mark pin 1, and then connect pins 1, 7, 10 and 16 to the +5 V bus. Connect pins 3, 4, 5, 6 and 8 to ground.

74LS00—Install a 14-pin socket at proto area U27. Again, mark pin 1 on the bottom of the board. Wire pins 1, 2, 9, 10, 12, 13 and 14 to +5. Wire pin 7 to ground.

74LS74—Install a 14-pin socket to U28. Wire pins 1, 4, 10, 11, 12, 13 and 14 to +5. Wire pin 7 to ground. Wire pin 6 to pin 2.

74LS151—Install a 16-pin socket at U44. Wire pins 4, 7, 8 and 9 to ground. Install a 4.7k resistor from pin 11 to +5 V and another 4.7k resistor from pin 10 to the +5 V bus. While installing these resistors, use the lead of one to also wire pins 12, 13, 14, 15 and 16 to the +5 V bus.

Following the signal path of Fig. 1, run a wire from U44, pin 5, to U26, pin 2. Connect pin 12 of U26 to pin 4 of U27. Connect pin 3 of U28 to pin 11 of U26 and to pin 5 of U27. Break the foil connections inside W5 and run a wire from the TX and RX ACIA inputs to pin 5 of U28. Finally, connect U27, pin 6, to U26, pin 9.

Now, pins 1, 2 and 3 of the 74LS151 (U44) must be connected to C4, C3 and C2.

C4—Locate U57 on the board. Pin 2 of U57 runs to a through-hole just below the IC (away from the keyboard). Run a wire from this through-hole to pin 1 of U44.

C3—Directly below U30 is a through-hole. Run a wire from it to

Address correspondence to Ronald W. Carr, 10A Morton St., Winthrop, ME 04364.

pin 2 of U44. You can verify with an ohmmeter that this through-hole is connected to pin 11 of U30.

C2—On the underside of the board, run a wire from pin 12 of U30 to pin 3 of U44. I could not find a convenient through-hole for this connection.

Wire the switch as shown in Fig. 1. The switch should have a center-off position in order to have three possible baud rates. Wire the switch with leads long enough to reach where you want it. Drill a hole and mount the switch. Install two .01 uF capacitors from +5 to ground near your work area. I installed one in U29 and another between U25 and U26 on the underside of the board.

Checkout

Reassemble the computer, plug in the new chips and turn on the power. With the switch in its center position the baud rate should be 300, and everything should work as usual. Load a short program, type SAVE and then LIST. As the program is listing, try the other two positions of the switch. In one it should list twice as fast, and in the other four times as fast. Next, try saving and reloading a program at 600 baud. Everything should work fine.

If some problem exists, first check your wiring. Unless the divider chain is shorted, problems can only be in the cassette load and save functions. You should be able to type in a program and then list it in save mode as above. The action of the computer as the switch is changed may give some indication of the problem. You can check the power to each IC with a voltmeter. Be sure to check the ground pins also. A floating ground causes just as much trouble as a missing +5 V. If further troubleshooting is necessary, use your ingenuity and whatever test equipment is available.

Operating

I have found the reliability at 600 baud very close to that at 300 baud. An error rarely occurs during a load —and this is with medium-grade audio tape for storage. Errors are made consistently if BASIC programs are recorded at 1200 baud. On lines longer than 24 characters, an error is made following each screen scroll because the system isn't fast enough to scroll and return in time to fetch the next character from the ACIA.

In BASIC in ROM versions, prints to the screen are vectored through locations 021A and 021B using an indirect jump. I tried changing these locations to jump to another routine without scrolls; this eliminated the consistent errors, but the reliability was not as good as 300 and 600 baud.

I use 600 baud for programs written in BASIC, and 1200 baud for long machine-language programs that load using a checksum for error detection. Any mistakes made during loading are easily corrected by rewinding the tape past the error and continuing with the load. ■

Parts List

SW1—SPDT switch with center off
U44—74LS151 multiplexer
U26—74LS161 counter
U27—74LS00 NAND gate
U28—74LS74 D-type flip-flop
(2)—4.7k resistors
(2)—16-pin IC sockets
(2)—14-pin IC sockets
(2)—.01 uF capacitors

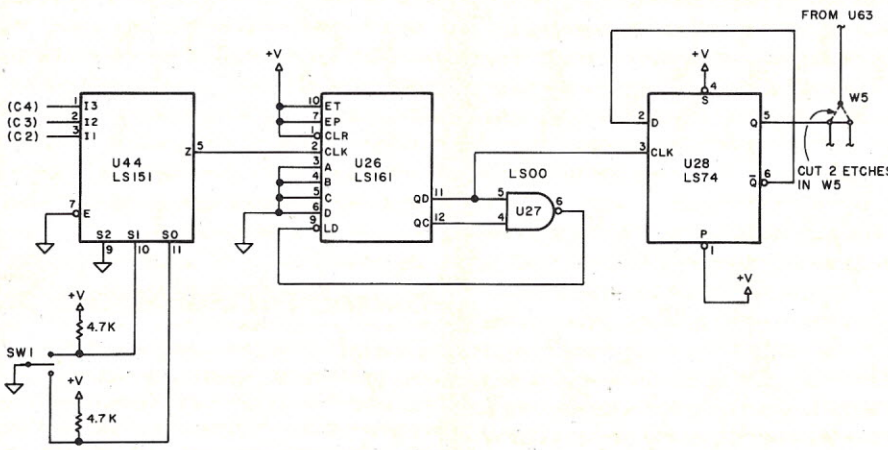


Fig. 1. Circuit to be constructed. IC designations (U26, etc.) correspond to those used by OSI in the C1P user's manual.

✓90

✓See List of Advertisers on page 210

This BASIC program is for authors, teachers, secretaries or anyone interested in the line, sentence, word or syllable content of a text, as well as its readability.

Minding Your P's and Q's

By Richard R. Parry

Your word-processing system enhances the cosmetic quality of text. Now all you need is some software to improve the writing.

The program described here helps do just that. It analyzes ordinary English text and computes the number of lines, sentences, words and syllables and the average word and sentence length. Then it gives a readability score.

You can also construct a concordance or frequency table of words. For example, this article contains 173 sentences, 3332 words and 5268 syllables, with average word and sentence lengths of 4.74 and 19.26, respectively. [Before editing—Eds.] The readability test indicates that this article should be easily comprehended by a person with an 11th grade reading level. In addition, a concordance table shows that the words "the," "of" and "to" appear 248, 130 and 85 times, respectively.

Armed with the result of the analysis, a writer can make his or her text more readable to the group he or she has targeted. For example, a good manager who is planning to distribute a paper describing a company-wide policy to all employees would not want the readability level to be that of a college student, since the reading level of the employees will most likely cover a wide range. The manager should strive to ensure that the level is sufficiently low to be easily comprehended by all employees.

Readability

Readability is the measurement of the comprehensibility of printed material. Educators have developed many methods for measuring readability. For example, Dr. Rudolf Flesch in *The Art of Plain Talk* describes a method based on the theory that the closer a word is to its root, the easier it will be understood. Therefore, prefixes and suffixes attached to words detract from readability. The average sentence length and use of proper nouns are other factors encompassed by the theory to arrive at an overall readability score.

The Fog Index is another means. It is based on the average sentence length and the percentage of "hard" words, which are uncommon words of three or more syllables. Other methods include the SRA, Botel, Dale-Chall and Fry readability tests.

The BASIC program shown in Listing 1 uses a readability method developed by Dr. Edward Fry. The technique is discussed by Dr. Fry in an article entitled, "A Readability Formula That Saves Time," which appeared in the April 1968 issue of the *Journal of Reading*. Ironically, the title is somewhat of a misnomer, since a graph and not a formula is used to arrive at a reading level. Variables used by the graph are sentences and syllables.

The original analysis method required a person to compute the average sentence length and the average number of syllables for three 100-word samples of the text. A better analysis could be derived if one analyzed the entire text; however, such a task for a human would indeed be tedious. Thanks to the computer, humanity is saved from this fate.

The results of the readability test yield one of 13 possible readability grade levels, ranging from first grade to college. A reading level of 12 represents the reading level of an average senior in high school.

A word of caution in interpreting reading scores: It is a mistake to think that a manuscript with a reading level of 12 is superior to one with a tenth grade level. In fact, one might argue that the converse is true, since the text with a tenth grade level can be comprehended by both a sophomore and a senior in high school, while the text with a readability level of 12 will be better understood by the senior. In actuality, each has its place. Prose should be written to meet the level of the group that will be reading it.

Richard R. Parry is an electrical engineer and programmer at the Fermi National Accelerator Laboratory, and teaches a course in Pascal at Elgin Community College in Illinois. His most recent article for Kilobaud Microcomputing *was "Math Teacher," in the April issue. He can be reached at 38 W. 255 Deerpath Road, Batavia, IL 60510.*

There are, of course, many factors affecting readability that the simple examination of the length of sentences and the number of syllables cannot take into account. The background of the reader and the interest that the text creates are important variables difficult to measure. In addition, readability formulas do not indicate conceptual difficulty due to unusual use of words, word order or the ratio of abstract to concrete terms. As Fry himself admits, the readability level computed using his graph is probably accurate to within a grade level.

Given these shortcomings, readability scores still serve as one important yardstick for measuring readability and aiding both an author to improve writing skills and an instructor to provide appropriate reading material.

Using the Program

The program begins by reminding the user that if a valid readability level is required, the minimum sample size of the text should not be less than 300 words. With this caveat understood, the user is then given the opportunity to indicate if the source of the text will be the computer's terminal or a text file.

Sample 1 shows how the user would input text from the terminal's keyboard. After indicating the source of the input, the user may also request a concordance table. Concordance tables come in a variety of forms. The table constructed by this program is basically a frequency table showing the number of times each unique word in the text occurs.

Assuming the table is not desired, the user responds with "N" for no. With these simple initialization steps complete, the user is prompted to input text. The text given by the user is analyzed by the computer one line at a time. When all text has been given, the user types a backslash character (\) on a new line to indicate the end of the text. The program then computes and displays the results of the analysis.

The results of the analysis for the text in Sample 1 shows that it consists of 34 lines, 19 sentences, 350 words and 567 syllables. In addition, the average word length is 4.78, and the average sentence is 18.42 words long. The readability test indicates that the text is written at a tenth grade reading level, which implies that a sophomore in high school should be able to comprehend the text with no difficulty.

Sample 2 shows the results of an analysis of the article you are now reading. The source of the input was a text file that was generated by a text editor and word processor. Using files to input text should be the more common means of supplying data to the program, since presumably the user has already prepared a letter or article as a file using a text editor and word processor. Most, if not all, word processors allow the user to direct the output to a file rather than to a terminal or printer. This file then serves as the input for the text analysis program. Note that even here, the end of text character is a backslash.

Since BASIC does not support a standard "end of file" function such as is found in Pascal or COBOL, it is necessary to edit the file to append the end of text character. Note that the values shown represent the specifications of the article as it was presented to the magazine's editor and possibly not as you see it in its present form.

After the user supplies the name of the file, the analysis begins. If the analysis is performed on a microcomputer, the user will have to exhibit some patience. Ideally, a program such as this should be written in assembly language or in a language that could be compiled. However, the program would then be much more limited. The execution time for the analysis varies linearly with the length of the text to be analyzed.

Of course, this time will vary greatly among computer systems, since the hardware as well as the version of BASIC will be major determining factors. On an SWTP computer with a Motorola 6800 running at 1 MHz with Technical Systems Consultant's version of BASIC, a 350 and 3000 word text analysis required 2.5 and 21 minutes, respectively.

The last example, Sample 3, shows how the program can be used to display a concordance table in addition to the text analysis. Assuming the text to be analyzed is a data file, the user starts the program as in Sample 2. However, when queried by the computer for a concordance table, the user responds with "Y."

The user now needs to approximate the number of unique words embodied in the text. This information is necessary since the words must be stored in an array and the computer must know how much memory to allocate. After the user re-

Listing 1. Complete BASIC program for the English text analysis and concordance program. No references are made to the REM statements; therefore, to conserve memory they may be removed.

```
100 REM ***********************************************************
110 REM *                                                         *
120 REM *      ENGLISH TEXT ANALYSIS AND CONCORDANCE PROGRAM      *
130 REM *                                                         *
140 REM *                          BY                             *
150 REM *                   RICHARD R. PARRY                      *
160 REM *                     JANUARY 1981                        *
170 REM *                                                         *
180 REM * THIS PROGRAM ACCEPTS ORDINARY ENGLISH TEXT AND COMPUTES *
190 REM * THE NUMBER OF: LINES, WORDS, SENTENCES, AND SYLLABLES   *
200 REM * AS WELL AS THE AVERAGE WORD AND SENTENCE LENGTH.        *
210 REM * A READABILITY SCORE IS ALSO GIVEN.  IF THE USER DESIRES *
220 REM * A CONCORDANCE TABLE MAY BE CONSTRUCTED.                 *
230 REM *                                                         *
240 REM *       LC.....LINE COUNTER                               *
250 REM *       SC.....SENTENCE COUNTER                           *
260 REM *       WC.....WORD COUNTER                               *
270 REM *       SM.....LETTER COUNTER                             *
280 REM *       SK.....SYLLABLE COUNTER                           *
290 REM *       NW.....UNIQUE WORD COUNTER                        *
300 REM *                                                         *
310 REM ***********************************************************
320 PRINT "NOTE!!! READABILITY TEST REQUIRES SAMPLE OF AT LEAST 300 WORDS"
330 PRINT "FOR BEST RESULTS."
340 PRINT : PRINT
350 INPUT "INPUT VIA KEYBOARD OR DISK (K/D)", QS$
360 IF QS$<>"K" AND QS$<>"D" THEN PRINT"K OR D PLEASE":GOTO350
370 INPUT "STORE WORDS FOR FREQUENCY ANALYSIS (Y/N)", ST$
380 IF ST$<>"Y" AND ST$<>"N" THEN PRINT"Y OR N PLEASE":GOTO370
390 IF ST$ = "N" THEN 430
400 INPUT "NUMBER OF DIFFERENT WORDS EXPECTED", DW
410 DIM W$(DW ): REM CONCORDANCE WORD ARRAY
420 DIM F(DW ) : REM FREQUENCY ARRAY OF EACH WORD
430 IF QS$= "D" THEN 470
440 PRINT "TERMINATE TEXT WITH A SINGLE LINE"
450 PRINT "CONSISTING OF ONLY A BACKSLASH (I.E. \)"
460 GOTO 490
470 INPUT "NAME OF FILE TO ANALYZE"; NF$
480 OPEN OLD NF$ AS 1
490 IF QS$ = "K" THEN INPUT LINE A$ ELSE INPUT LINE #1, A$ : PRINT "*";
500 IF LEN(A$) = 1 AND A$ = "\" THEN 1050 : REM CHECK FOR END OF TEXT
510 IF LEN(A$) = 0 THEN 490  : REM NULL INPUT STRING
520 A$ = A$ + " "            : REM APPEND A SPACE CHARACTER TO EOL
530 EN  = LEN(A$)            : REM INITIALIZE LENGTH OF LINE
540 LC  =LC  + 1             : REM INCREMENT LINE COUNTER
550 CO  = 0                  : REM INITIALIZE WORD LENGTH COUNTER
560 WF  = 0                  : REM INITIALIZE WORD FLAG
570 REM ****************************************************************
580 REM * ANALYZE INPUT LINE CHARACTER BY CHARACTER CHECKING FOR A WORD *
590 REM * OR SENTENCE                                                  *
600 REM ****************************************************************
610 FOR I  = 1 TO EN
620   CH$ = MID$(A$, I , 1)
630   REM ***********************************************************
640   REM * TEST FOR SENTENCE WHICH IS A . OR ? OR ! PRECEEDED        *
650   REM * BY A WORD WITH LENGTH >1 CHARACTER AND FOLLOWED BY A SPACE *
660   REM ***********************************************************
670   ES  = 0
680   NC$ = MID$(A$, I +1, 1)   : REM SAVE NEXT CHARACTER
690   IF CH$ = "." OR CH$ = "?" OR CH$ = "!" THEN ES  = 1
700   IF ES  = 1 AND (CO >1 OR PC$ = CHR$(34)) AND NC$ = " " THEN SC  = SC  +1
710   REM *******************************************************
720   REM * TEST FOR A WORD WHICH IS DETERMINED BY THE PRESENCE OF  *
730   REM *   SPACE .  ,  ;  :  ?  !    "   (    )                  *
740   REM *******************************************************
750   IF CH$ = " " OR CH$ = CHR$(34) OR CH$ = "(" OR CH$ = ")" THEN 800
760   IF CH$=":" OR CH$=";" OR CH$="?" OR CH$="!" THEN 800
770   IF CH$="," OR CH$="."  THEN 790
780   CO  = CO  + 1 : WF  = 1 : GOTO 990
790   IF MID$(A$, I +1, 1) <> " " THEN CO  = CO  + 1 : GOTO 1000
800   IF WF  = 0 THEN 990    : REM LETTER MUST PRECEED PRESENT LETTER
810   SM  = SM  + CO         : REM INCREMENT LETTER COUNTER
820   WC  = WC  + 1          : REM INCREMENT WORD COUNTER
830   WF  = 0                : REM RESET WORD FLAG
840   WR$ = MID$(A$, I -CO , CO )
850   GOSUB 2170             : REM ANALYZE WORD FOR SYLLABLES
860   SK  = SK  + SY         : REM INCREMENT TOTAL SYLLABLE COUNTER
870   IF ST$ = "N" THEN 980 : REM SKIP CONCORDANCE STORING ROUTINE
880   REM ***********************************************
890   REM * SEARCH TABLE FOR WORD.  IF NEW WORD ADD TO LIST *
900   REM * IF OLD WORD, INCREMENT FREQUENCY COUNTER        *
910   REM ***********************************************
920   FOR J  = 1 TO NW
930     IF W$(J ) = WR$ THEN F(J ) = F(J ) + 1 : GOTO 980
940   NEXT J
950   NW  = NW  + 1          : REM INCREMENT "NEW" WORD COUNTER
960   W$(NW ) = WR$          : REM STORE NEW WORD
970   F(NW ) = 1             : REM NEW WORD. THERE IS ONLY ONE
980   CO  = 0                : REM ZERO WORD CHARACTER COUNTER
990   PC$ = CH$              : REM SAVE PREVIOUS CHARACTER
1000 NEXT I
1010 GOTO 490
1020 REM ************************************************************
1030 REM * TEXT ANALYSIS COMPLETE. DISPLAY RESULTS SPECIFIED BY USER *
1040 REM ************************************************************
1050 IF ST$ = "N" THEN 1560
1060 PRINT : PRINT : PRINT
1070 PRINT "*** ANALYSIS OPTIONS ***" : PRINT
1080 PRINT "1. FREQUENCY SORT"
1090 PRINT "2. ALPHABETIC SORT"
```

More

sponds to the question and supplies the name of the text file, the computer begins the analysis.

Two points are worthy of note in reference to using the concordance feature of the program. The first is the amount of time required to execute the program. While a large-scale computer can analyze long manuscripts in moments, a microcomputer using a BASIC interpreter requires much more time. For example, a 1100 word manuscript that required nine minutes to analyze without a concordance table required 21 minutes when the concordance table was requested.

Second, the construction of a concordance table requires a good deal of memory since all words that make up the table must be simultaneously in memory. Most microcomputer systems with 16K of memory available for variables should be able to construct a table of 900 words. For most articles this should be adequate since an average article of 3000 words will most likely have less than 900 unique words.

When the analysis is complete, the user is shown a menu from which he or she may indicate the information desired and the format the data is to take. You can request that the concordance table be displayed in alphabetical order, in order of highest frequency, or merely unsorted. Sorting is another time-consuming process that the user should be aware of if a microcomputer is being used. Sorting 400 words on the computer system mentioned above requires 25 minutes. Sample 3 shows the table sorted according to frequency.

Notice that the word "the" is considered to be different from the word "The." It is possible to convert all uppercase characters to lowercase so that this apparent oddity will not occur. However, converting the case of the characters leads to an even more undesirable situation. Names such as "John Q. Public" and "Illinois" would then be shown in the table as "john q. public" and "illinois," which would be even more incorrect.

About the Program

After many years of programming, one thing has become apparent to me: most programming projects that at first seem innocuous are anything but. The present program is no exception. Parsing a line of text for words and sentences at first glance

See List of Advertisers on page 210

Listing 1 continued.

```
1100 PRINT "3. NO SORT"
1110 PRINT "4. STATISTICS ONLY"
1120 PRINT "5. END"
1130 PRINT : PRINT
1140 INPUT "SELECT OPTION (1,2,3,4,5)"; QS
1150 IF QS <1 OR QS >5 THEN 1060
1160 ON QS  GOTO 1200, 1340, 1480, 1560, 2090
1170 REM ****************************************
1180 REM * BUBBLE SORT ACCORDING TO FREQUENCY *
1190 REM ****************************************
1200 PRINT : PRINT "*** SORTING IN PROGRESS ****"
1210 FL  = 0
1220 FOR K  = 1 TO NW -1
1230   IF F(K ) >= F(K +1) THEN 1280
1240   SW  = F(K )            : S$ = W$(K )
1250   F(K )  = F(K +1)       : W$(K )  = W$(K +1)
1260   F(K +1)  = SW          : W$(K +1)  = S$
1270   FL  = 1
1280 NEXT K
1290 IF FL  = 1 THEN 1210
1300 GOTO 1480 : REM SORT COMPLETE, DISPLAY TABLE
1310 REM ****************************************
1320 REM * BUBBLE SORT ACCORDING TO ALPHABETIC*
1330 REM ****************************************
1340 PRINT : PRINT "*** SORTING IN PROGRESS ****"
1350 FL  = 0
1360 FOR K  = 1 TO NW -1
1370   IF W$(K )  <= W$(K +1)  THEN 1420
1380   SW  = F(K )            : S$ = W$(K )
1390   F(K )  = F(K +1)       : W$(K )  = W$(K +1)
1400   F(K +1) = SW           : W$(K +1) = S$
1410   FL  = 1
1420 NEXT K
1430 IF FL  = 1 THEN 1350
1440 GOTO 1480 : REM SORT COMPLETE, DISPLAY TABLE
1450 REM ****************************************
1460 REM * PRINT FREQUENCY DISTRIBUTION TABLE *
1470 REM ****************************************
1480 PRINT : PRINT : PRINT TAB(7) ; "*** CONCORDANCE TABLE ***"
1490 PRINT : PRINT "WORD #              WORD              FREQUENCY"
1500 FOR K  = 1 TO NW
1510   PRINT K , W$(K ) , F(K )
1520 NEXT K
1530 REM ************************************
1540 REM * PRINT TEXT STATISTICAL ANALYSIS *
1550 REM ************************************
```

More

seems easy. Sentences could be counted by merely noting the number of periods. Words could be counted by counting the number of spaces. However, the number of anomalies in the English language requires more sophisticated algorithms for an accurate word and sentence count. The program contains three major algorithms that are worthy of special note and explanation—the sentence, word and syllable algorithms.

Before discussing the specifics of the algorithms, I should mention that I had to make a trade-off early in the design phase of the sentence algorithm. Either the algorithm could be made very accurate for nonfiction, or mediocre for prose in general. I chose the former, because of the way periods, question marks and other sentence-terminating characters are used in fiction and nonfiction.

Typically in nonfiction, a period is preceded by a word and followed by a space. This is not true in fiction. Fiction is inundated with the quotations of characters in the story. The quotation marks surrounding the remarks of the story's characters break the sentence rule mentioned. Targeting the program for nonfiction writers

seems a reasonable trade-off, since I suspect that this program will rarely, if ever, be used by fiction writers.

The first approximation at a sentence counter algorithm would, of course, count periods. However, if a sentence contained the name "John Q. Public," the period after the letter "Q" would cause a sentence miscount. This inaccuracy is overcome by checking that more than one character precedes the period. It is true that one could think of cases in which the algorithm fails, such as the sentence, "He is a better chess player than I." Here the period after the word "I" would not denote a sentence, because the sentence ends with a single character word. However, after studying hundreds of sentences in ordinary text, I rarely found sentences that fooled the algorithm.

Our sentence algorithm must be further enhanced before we can be confident in its accuracy. If, for example, some text contained a number such as 47.12, the decimal point (period) following the first two digits would be construed as indicating the end of a sentence, which, of course, it is not. This dilemma is solved by

Listing 1 continued.

```
1560 SK  = INT( SK  * .99 + .5) : REM INACCURATE SYLLABLE ALGORITHM REDUCE 1
1570 PRINT : PRINT : PRINT
1580 PRINT "*** TEXT ANALYSIS STATISTICS ***" : PRINT
1590 PRINT "NUMBER OF LINES "; LC
1600 PRINT "NUMBER OF SENTENCES "; SC
1610 PRINT "NUMBER OF WORDS "; WC
1620 PRINT "NUMBER OF SYLLABLES ";SK
1630 PRINT "AVERAGE WORD LENTH "; INT( SM/WC  * 100 + .5) / 100
1640 PRINT "AVERAGE SENTENCE LENGTH "; INT( WC  / SC  * 100 + .5) / 100
1650 IF WC  > 300 THEN 1740
1660 PRINT "*** FOR MOST ACCURATE RESULTS, THE READABILITY TEST REQUIRES ***"
1670 PRINT "*** A SAMPLE OF GREATER THAN 300 WORDS.  THEREFORE, GRADE    ***"
1680 PRINT "*** LEVEL GIVEN BELOW MAY BE INACCURATE.                     ***"
1690 REM **********************************************************
1700 REM * BASED ON THE NUMBER OF SYLLABLES AND SENTENCES DISCOVERED, FIND *
1710 REM * THE GRADE LEVEL AT WHICH THE TEXT WAS WRITTEN.  FIRST NORMALIZE *
1720 REM * TO 100 WORD SAMPLE TO USE DR. FRY'S GRAPH.                      *
1730 REM **********************************************************
1740 NS = SC  * 100.0 / WC  : REM NORMALIZE # OF SENTENCES TO 100 WORD SAMPLE
1750 NY = SK  * 100.0 / WC  : REM NORMALIZE # OF SYLLABLES TO 100 WORD SAMPLE
1760 DIM S( 12 ) , Y( 12 )       : REM SENTENCE & SYLLABLE VALUES FOR FRY'S GRAPH
1770 IF NY < 108 OR NY > 172 OR NS < 3.6 OR NS > 25 THEN 1780 ELSE 1800
1780 PRINT "*** DATA GIVEN IS OUTSIDE VALID RANGE FOR READABILITY TEST   ***"
1790 GOTO 2080
1800 S( 12 ) = .283 * NY -41.7 : Y( 12 ) = 3.46 * NS + 147.7
1810 S( 11 ) = .191 * NY -25.8 : Y( 11 ) = 5.17 * NS + 135.4
1820 S( 10 ) = .177 * NY -22.7 : Y( 10 ) = 5.58 * NS + 128.6
1830 S( 9 )  = .184 * NY -22.8 : Y( 9 )  = 5.36 * NS + 124.2
1840 S( 8 )  = .158 * NY -17.6 : Y( 8 )  = 6.20 * NS + 112.0
1850 S( 7 )  = .137 * NY -13.6 : Y( 7 )  = 7.18 * NS +  99.8
1860 S( 6 )  = .170 * NY -16.0 : Y( 6 )  = 5.77 * NS +  94.3
1870 S( 5 )  = .161 * NY -13.1 : Y( 5 )  = 6.01 * NS +  82.6
1880 S( 4 )  = .173 * NY -13.6 : Y( 4 )  = 5.64 * NS +  79.9
1890 S( 3 )  = .222 * NY -18.6 : Y( 3 )  = 4.37 * NS +  84.7
1900 S( 2 )  = .305 * NY -25.7 : Y( 2 )  = 3.21 * NS +  85.2
1910 S( 1 )  = .469 * NY -41.5 : Y( 1 )  = 2.08 * NS +  89.5
1920 IF NS< S( 12 ) AND NY> Y( 12 ) THEN GR  = 13
1930 IF NS> S( 12 ) AND NS<= S( 11 ) AND NY< Y( 12 ) AND NY>= Y( 11 ) THEN GR  = 12
1940 IF NS> S( 11 ) AND NS<= S( 10 ) AND NY< Y( 11 ) AND NY>= Y( 10 ) THEN GR  = 11
1950 IF NS> S( 10 ) AND NS<= S( 9 )  AND NY< Y( 10 ) AND NY>= Y( 9 )  THEN GR  = 10
1960 IF NS> S( 9 )  AND NS<= S( 8 )  AND NY< Y( 9 )  AND NY>= Y( 8 )  THEN GR  = 9
1970 IF NS> S( 8 )  AND NS<= S( 7 )  AND NY< Y( 8 )  AND NY>= Y( 7 )  THEN GR  = 8
1980 IF NS> S( 7 )  AND NS<= S( 6 )  AND NY< Y( 7 )  AND NY>= Y( 6 )  THEN GR  = 7
1990 IF NS> S( 6 )  AND NS<= S( 5 )  AND NY< Y( 6 )  AND NY>= Y( 5 )  THEN GR  = 6
```

More

Listing 1 continued.

```
2000 IF NS> S(5)  AND NS<= S(4)  AND NY< Y(5)  AND NY>= Y(4)  THEN GR  = 5
2010 IF NS> S(4)  AND NS<= S(3)  AND NY< Y(4)  AND NY>= Y(3)  THEN GR  = 4
2020 IF NS> S(3)  AND NS<= S(2)  AND NY< Y(3)  AND NY>= Y(2)  THEN GR  = 3
2030 IF NS> S(2)  AND NS<= S(1)  AND NY< Y(2)  AND NY>= Y(1)  THEN GR  = 2
2040 IF NS> S(1) AND NY< Y(1) THEN GR  = 1
2050 PRINT "BASED ON NUMBER OF SENTENCES AND SYLLABLES "
2060 PRINT "     YOUR READABILITY GRADE LEVEL IS ";
2070 IF GR  = 13 THEN PRINT "COLLEGE" ELSE PRINT GR
2080 IF ST$ = "Y" THEN 1060 : REM RETURN TO MENU
2090 END
2100 REM ***********************************************************
2110 REM * COUNT SYLLABLES IN WORD.  A SYLLABLE COUNTER IS INCREMENTED  *
2120 REM * EACH TIME A VOWEL OR Y IS FOUND.  HOWEVER, THEY MUST BE      *
2130 REM * SEPARATED BY A DIFFERENT CHARACTER.  IN ADDITION, EACH WORD  *
2140 REM * MUST HAVE AT LEAST ONE SYLLABLE. IF THE WORD ENDS IN "E"     *
2150 REM * THE COUNTER IS DECREMENTED.                                  *
2160 REM ***********************************************************
2170 SY  = 0                : REM INITIALIZE SYLLABLE COUNTER
2180 SF  = 0                : REM INITIALIZE SYLLABLE FLAG
2190 FOR L  = 1 TO CO
2200   CR$ = MID$(WR$,L ,1)
2210   IF CR$="a" OR CR$="e" OR CR$="i" THEN 2260
2220   IF CR$="o" OR CR$="u" OR CR$="y" THEN 2260
2230   IF CR$="A" OR CR$="E" OR CR$="I" THEN 2260
2240   IF CR$="O" OR CR$="U" OR CR$="Y" THEN 2260
2250   SF  = 0  : GOTO 2290
2260   IF SF  = 1 THEN 2290
2270   SY  = SY  + 1          : REM INCREMENT SYLLABLE COUNTER
2280   SF  = 1                : REM SET SYLLABLE FLAG
2290 NEXT L
2300 IF CR$ = "E" OR CR$ = "e" THEN SY  = SY  - 1
2310 IF SY  < 1 THEN SY  = 1: REM 1 SYLLABLE MINIMUM
2320 RETURN
```

Sample 1. This sample run shows how text which is to be analyzed is given by the user from the terminal's keyboard. The output indicates the number of lines, sentences, words and syllables. The average word and sentence length is also displayed in addition to the readability level. The text is an excerpt from Collier's Encyclopedia.

```
NOTE!!! READABILITY TEST REQUIRES SAMPLE OF AT LEAST 300 WORDS
FOR BEST RESULTS.


INPUT VIA KEYBOARD OR DISK (K/D)? K
STORE WORDS FOR FREQUENCY ANALYSIS (Y/N)? N
TERMINATE TEXT WITH A SINGLE LINE
CONSISTING OF ONLY A BACKSLASH (I.E. \)
?     The whale is an aquatic mammal that breathes air but cannot
? survive on dry land.  It is the largest known mammal and lives in the
? ocean where it must surface at intervals to obtain air.  A whale
? bears a superficial resemblance to the fishes in external form and
? is streamlined for an aquatic life.  Its general body form is round,
? elongate, and tapered posteriorly to a pair broad, flattened,
? horizontal flukes.  The flukes, which make up the tail, are notched
? in the middle, have no bony support, and serve as the propelling
? force that drives the animal through the water.  They are made up of
? cartilaginous tissue and are placed in a horizontal position.  The
? forelimbs or flippers of the whale correspond to the arms and hands
? of land mammals.  The parts that correspond to the fingers and hands
? are encased in a cartilaginous covering of skin and are shaped like
? a paddle.  They serve as stabilizers to maintain an even keel for the
? animal.  The whale has no hind legs although remnants of a pelvis and
? limb bones are found in the fleshy parts of some species.
?     Most modern whaling is carried out on the high seas,
? or at least beyond the three-mile limit.
? Whales generally are available to whalemen of any nation,
? and floating-factory ships operating on a pelagic basis
? take more whales than the more numerous shore stations.
? Consequently, the regulation of whaling for the conservation
? of the stock must be done mainly by international law on the
? basis of treaties.
?     Whaling regulations have been in existence for centuries.
? However, they have not been completely successful in curtailing
? overhunting, and the depletion of whale populations has been
? quite rapid.  The first species to be depleted was the humpback.
? Then the blue whale, the most desirable species, showed a
? population decrease.  For example, the Antartic population of blue
? whale averaged about 14,500 in the 1930's.  By the early 1960's
? population figures had dropped to about 2,000, and in 1968 it was
? estimated that only about 600 of the animals were left in the
? vast ocean surrounding the Antartic.
\


*** TEXT ANALYSIS STATISTICS ***

NUMBER OF LINES  34
NUMBER OF SENTENCES  19
NUMBER OF WORDS  350
NUMBER OF SYLLABLES  551
AVERAGE WORD LENGTH  4.78
AVERAGE SENTENCE LENGTH  18.42
BASED ON NUMBER OF SENTENCES AND SYLLABLES
     YOUR READABILITY GRADE LEVEL IS  10
```

checking for a space after the period. However, even here we can get into difficulty if the period is the last character in the line. Therefore, a space character is appended to each line of text that is read to ensure an accurate sentence count using this algorithm. Adding a space to the end of the line also simplifies the word algorithm which will be discussed shortly.

Lastly, since sentences may terminate with a question mark or exclamation point as well as a period, these characters are also encompassed by the algorithm and treated in the same manner as a period.

The word algorithm is a real bucket of worms. It must properly detect and extract a word from a line and delete excess punctuation marks or characters, such as spaces, periods, commas, question marks and others that appear in normal text. It is true that words could be accurately counted by merely counting spaces or groups of spaces between nonspace characters. However, to accurately construct a concordance table, the word must be extracted from punctuation marks that may surround it. For example, if the word "standard" appeared in text with quotation marks on either side of it, and later appeared without quotes around it, we would still consider them to be the same word. We must, therefore, force the program to also consider them the same word. This necessitates removal of punctuation marks, with demands a more complex algorithm.

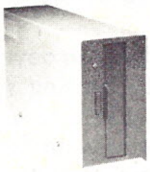
The nine punctuation marks that appear commonly in English text and must be detected and treated properly are colons, semicolons, quotation marks, question marks, exclamation points, right and left parentheses, commas and periods. The first seven are removed whether they appear at the beginning or end of a word. Therefore, the word "standard" in quotes would be displayed in the concordance table without quotes.

The comma and period are treated differently, since in the case of numbers, we would want to save them as part of the number. They are thus removed only if followed by a space and preceded by a character other than the nine punctuation marks mentioned. In this way, a number such as 2.4 would be labelled a single word, as would the dollar amount $2,134.98.

In addition, an ellipsis (. . .) would not be considered a word. Nor would it be considered three sentences ac-

WE WILL NOT BE UNDERSOLD

DISK DRIVES



FOR TRS-80* Model I

CCI-100	5¼", 40 Track (102K)	**$314**
CCI-280	5¼", 80 Track (204K)	**$429**

ADD-ON DRIVES FOR ZENITH Z-89

CCI-189	5¼", 40 Track (102K)	**$394**
CCI-289	5¼", 80 Track (204K)	**$499**
Z-87	Dual 5¼" system	**$995**

External card edge and power supply included. 90 day warranty/one year on power supply.

RAW DRIVES 8" SHUGART 801R **$425**
5¼" TEAC or TANDON **$ CALL** POWER SUPPLIES **$ CALL**

MORROW DESIGNS/THINKER TOYS™

DISCUS 2D	1 DRIVE	**$ 938**	2 DRIVE	**$1635**
DISCUS 2+2	1 DRIVE	**$1259**	2 DRIVE	**$2245**
DISCUS Hard Disk	M26	**$3990**	M10	**$2999**

DISKETTES — Box of 10

5¼"	**Scotch**	**$35**	**Maxell**	**$40**	**BASF/Verbatim**	**$26.95**
8"	**Scotch**	**$50**	**Maxell**	**$55**	**BASF/Verbatim**	**$36.00**

PLASTIC FILE BOX-Holds 50 5¼" diskettes **$19.00**
PLASTIC LIBRARY CASE 5¼" **$3.00** 8" **$ 4.00**
SCOTCH HEAD CLEANING DISKETTE **$25.00**
FLOPPY SAVER **$11.95** RINGS **$ 6.95**

SYSTEM SPECIAL

Apple II Plus 48K w/drive and controller. Epson MX-80 printer and interface. SUP-R Mod RF Modulator: List **$2965** You Pay **$2299**

COMPUTERS/TERMINALS

ARCHIVES	64K, 2-Drives, 77 Track			**$5495**
ALTOS	ACS8000 Series			**$ CALL**
TRS-80*	II-64K	**$3499**	III-16K	**$ 899**
TRS-80*	Expansion Interface			**$ 299**
ZENITH	48K, all-in-one computer			**$2200**
ZENITH	Z-19			**$ 735**
TELEVIDEO	920C	**$ CALL**	950	**$ CALL**
IBM	3101 Display Terminal			**$1189**
ATARI	400	**$ 479**	800	**$ 795**
MATTEL	INTELLIVISION			**$ 229**
APPLE PERIPHERALS				**$ CALL**

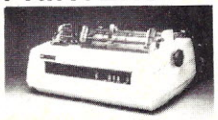
MONITORS

APF	9" B & W TVM-10			**$120**
BELL & HOWELL	9" B & W BHD911			**$199**
LEEDEX	12" B & W	**$ 129**	13" Color	**$369**
SANYO	9" B & W VM4509			**$155**
SANYO	12" B & W DM5012			**$226**
SANYO	12" Green Screen DM5112			**$238**
SANYO	13" Color DMC6013			**$416**
ZENITH	13" Color			**$349**

TELECOMMUNICATIONS

LIVERMORE STAR MODEM 2-year guarantee	**$125**
UNIVERSAL DATA SYSTEMS UDS-103	**$179**
D-CAT HARD WIRED DIRECT MODEM	**$189**
AUTO-CAT Auto Answer, Direct Connect Modem	**$249**
D.C. HAYES MICRO-MODEM	**$329**

PRINTERS



NEC Spinwriter
Letter Quality High Speed Printer

R.O.	**$2395**
R.O. with tractor feed	**$2595**
KSR with tractor feed	**$2895**

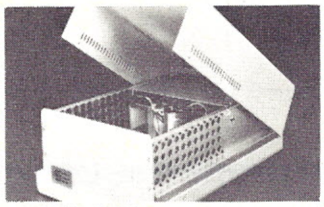
C.ITOH	Starwriter **$1695**		Starwriter II	**$1895**
EPSON	MX-80 **$CALL**		MX-70	**$CALL**
PAPER TIGER				
IDS 445	Graphics & 2K buffer			**$ 699**
IDS 460	Graphics & 2k buffer			**$1050**
IDS 560	Graphics			**$1450**
ANADEX	DP-8000 **$ 849**		DP-9500/01	**$1345**
OKIDATA				
Microline 80	Friction & pin feed			**$ 420**
Microline 80	Friction, and pin & tractor feed			**$ 520**
Microline 82	Friction & pin feed feed			**$ 620**
Microline 83	120 cps, uses up to 15" paper			**$ 849**
CENTRONICS	730 **$ 595**	799 **$ 969**	737	**$ 749**
TI-810				

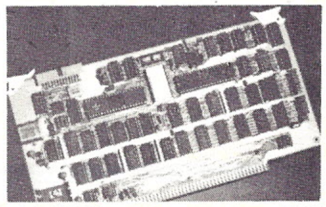
TRS-80* software, compressed print & vert. form control **$1865**

16K RAM KITS

2 for $37 **$19**
200 ns for TRS-80*, Apple II, (specify): **Jumpers** **$2.50**

S-100 CALIFORNIA COMPUTER SYSTEMS





MAINFRAME	Model 2200A	**$349**
Z80 CPU	Model 2810	**$269**
MOTHER BOARD	Model 2501	**$106**
16K STATIC RAM, 200ns	Model 2116C	**$309**
32K STATIC RAM, 200ns	Model 2032C	**$619**
64K DYNAMIC RAM	Model 2065C	**$599**
FLOPPY DISC CONTROLLER	Model 2422A	**$359**
EXTENDER BOARD	Model 2520K	**$ 52**
2P + 2S I/O	Model 2718A	**$309**

APPLE ACCESSORIES AND SOFTWARE

VISICALC	**$120.00**
DB MASTER	**$159.00**
Z-80 SOFTCARD	**$269.00**
VIDEX BOARD	**$259.00**
16K CARD	**$169.00**
APPLE JOYSTICK	**$ 49.00**
SUP-R MOD	**$ 29.00**
CCS CARDS	**$ CALL**
GALAXIAN	**$ 22.95**
SPACE ALBUM	**$ 35.00**
ASTEROIDS	**$ 17.95**
FLIGHT SIMULATOR	**$ 29.00**
WIZARD & PRINCESS	**$ 28.00**
SARGON 2	**$ 29.00**
HI-RES FOOTBALL	**$ 35.00**
MYSTERY HOUSE	**$ 21.00**

For fast delivery, send certified checks, money orders or call to arrange direct bank wire transfers. Personal or company checks require two to three weeks to clear. All prices are **mail order only** and are subject to change without notice. **Call for shipping charges.**

DEALER (NATIONAL/INTERNATIONAL) INQUIRIES INVITED **Send for FREE Catalogue**

The CPU SHOP

256

TO ORDER CALL TOLL FREE 1-800-343-6522
TWX: 710-348-1796 Massachusetts Residents call 617/242-3361

5 Dexter Row, Dept. K06M
Charlestown, Massachusetts 02129
Hours 10AM-6PM (EST) Mon.-Fri. (Sat. till 5)

Technical Information call 617/242-3361
Massachusetts Residents add 5% Sales Tax
Tandy Corporation Trademark/® Digital Research

master charge THE INTERBANK CARD
VISA®

See List of Advertisers on page 210

Sample 2. In this example, we see the analysis of the text file which was used to write this article. Instead of typing the text, as shown in Sample 1, a previously created text file is called. An asterisk is printed each time a line is read from the text file to provide the user with feedback as to the progress of the analysis.

```
RUN

NOTE!!! READABILITY TEST REQUIRES SAMPLE OF AT LEAST 300 WORDS
FOR BEST RESULTS.


INPUT VIA KEYBOARD OR DISK (K/D)? D
STORE WORDS FOR FREQUENCY ANALYSIS (Y/N)? N
NAME OF FILE TO ANALYZE? ARTICLE.TXT

**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
*************************************************************************


*** TEXT ANALYSIS STATISTICS ***

NUMBER OF LINES  326
NUMBER OF SENTENCES  173
NUMBER OF WORDS  3332
NUMBER OF SYLLABLES  5268
AVERAGE WORD LENTH  4.74
AVERAGE SENTENCE LENGTH  19.26

BASED ON NUMBER OF SENTENCES AND SYLLABLES
     YOUR READABILITY GRADE LEVEL IS  11
```

Sample 3. This last sample shows the ability of the program to construct a concordance table according to one of two different sorting keys. Note that even though there are 350 words, there are only 188 unique words. The text analyzed is identical to that shown in Sample 1.

```
RUN

NOTE!!! READABILITY TEST REQUIRES SAMPLE OF AT LEAST 300 WORDS
FOR BEST RESULTS.


INPUT VIA KEYBOARD OR DISK (K/D)? D
STORE WORDS FOR FREQUENCY ANALYSIS (Y/N)? Y
NUMBER OF DIFFERENT WORDS EXPECTED? 210
NAME OF FILE TO ANALYZE? WHALE.TXT
***********************************

*** ANALYSIS OPTIONS ***

1. FREQUENCY SORT
2. ALPHABETIC SORT
3. NO SORT
4. STATISTICS ONLY
5. END


SELECT OPTION (1,2,3,4,5)? 1

*** SORT IN PROGRESS ***


             *** CONCORDANCE TABLE ***

WORD #                 WORD            FREQUENCY
1                      the                30
2                      of                 13
3                      and                12
4                      in                 11
5                      to                 9

6                      a                  8
7                      whale              7
8                      are                7
9                      The                6
10                     is                 5

11                     that               4
12                     on                 4
 .                      .                 .
```

More

cording to the sentence algorithm previously discussed. Once again, after studying nearly 1000 words appearing in ordinary text, the algorithm has proven to be virtually perfect.

Our last task requires the development of a means of counting syllables. If you want to make a programmer laugh, tell him you want him to write a program to detect and count "a segment of speech typically produced with a single pulse of air pressure from the lungs, and consisting of a center of relatively great sonority with or without one or more accompanying sounds of relatively less sonority." That is the definition of a syllable according to the *Random House Dictionary of the English Language*. Developing a means of precisely counting syllables by a machine is impossible unless an exhaustive search is made of every word known in the English language.

Despite this somewhat nebulous definition, we can come surprisingly close. By counting every occurrence of the five vowels and the pseudo vowel Y, and checking that they are separated by consonants, we are in most cases correct. For example, the word "beads" is a single syllable word even though there are two vowels, because the vowels are not separated by a consonant. The algorithm's accuracy can further be enhanced by decreasing the syllable count if the word being analyzed ends with the letter "E." This is referred to as the silent "E" by reading teachers. Adding this requirement causes words such as "force," "space" and "nice" to be properly labelled as single-syllable words.

One last additional criterion must be met. All words must have at least one syllable. Therefore, the letter "Q" in John Q. Public would be a single-syllable word, as would the number 37.

To test the accuracy of the algorithm, an examination of 300 "high-frequency words" and their variants was made. These words are referred to as high-frequency words, since they make up 65 percent of all words written in English. In fact, finding a sentence that does not contain at least one of the first ten words of the list is next to impossible. This research showed that the syllable algorithm yielded a count slightly too high.

An additional study of 3000 words of running text containing nearly 5000 syllables was made to further

verify the accuracy of the syllable algorithm (counting several thousand syllables has to rank as one of the more boring tasks of life). The results of this second study showed the algorithm to be virtually perfect. A compromise was struck between the results found for each study. The final value displayed as the syllable count is reduced by 1 percent to yield an exceptionally accurate syllable count.

Conclusion

A text analysis program is the next logical step after the word processor toward better writing. Readability scores are not a sure measure of effective communication, but used in conjunction with other techniques, they serve as the basis for unbiased comparison.

Several times throughout the article I have mentioned that the execution time of the analysis is slow due mainly to the use of a microcomputer and a BASIC interpreter. However, you should remember that the computer is doing the work and not you. Go have a cup of coffee, relax and think of your next article while you are waiting. ■

Sample 3 continued.

```
 .                    .                  .
 .                    .                  .
 .                    .                  .
 .                    .                  .
190                  had                 1
191                  dropped             1
192                  2,000               1
193                  1968                1
194                  estimated           1
195                  only                1
196                  600                 1
197                  animals             1
198                  were                1
199                  left                1
200                  vast                1
201                  surrounding         1


*** TEXT ANALYSIS STATISTICS ***

NUMBER OF LINES  34
NUMBER OF SENTENCES  19
NUMBER OF WORDS  350
NUMBER OF SYLLABLES  551
AVERAGE WORD LENGTH  4.78
AVERAGE SENTENCE LENGTH  18.42
BASED ON NUMBER OF SENTENCES AND SYLLABLES
     YOUR READABILITY GRADE LEVEL IS  10


*** ANALYSIS OPTIONS ***

1. FREQUENCY SORT
2. ALPHABETIC SORT
3. NO SORT
4. STATISTICS ONLY
5. END

SELECT OPTION (1,2,3,4,5)? 5
```

Operate your micro in even the most hostile environs with a TASA touch-activated keyboard.

Light-Fingered Computing

By Douglas H. Haden

Photo 1. The TASA Model 55 Keyboard The fully-sealed polycarbonate keyboard contains the electronics for 55 microproximity sensor pads, 128-position ASCII encoding, parity and strobe. The four-color layout, sizing and choice of nomenclature result in a friendly-appearing keyboard.

I became interested in the TASA keyboard while seeking some alternative to spending many hours a day hunched over a typewriter. My dream was a keyboard that I could use while I twisted and turned in an overstuffed chair. The TASA keyboard weighed about 20 ounces and could be used at a desk on its optional stand or laid in my lap.

It has met my needs well. I've dropped it, spilled coffee on it, sat on it, and generally abused it and have had no problems to date.

Manufactured by Touch Activated Switch Arrays and available from KLM Marketing, PO Box 1237, Los Altos, CA 94022, for $75 plus $5 postage and handling, the TASA model 55 keyboard is fully ASCII encoded, with parity and strobe, and is sealed in a 6.25 × 15 inch monolith less than a third of an inch thick. Photo 1 shows the TASA 55 keyboard; Fig. 1 shows the key nomenclature and the encoded output.

The keys are activated by finger pressure on touch-pads on the four-color keyboard face, using neither contact closure nor mechanical motion. When a given key is activated, its encoded binary output (along with parity and strobe) is available at the output connector. All touch-pads, except the shift, shift-lock and control keys, feature two-key rollover; if more than one key is activated, no output is generated.

Interface Circuits

I interfaced my TASA keyboard to a Sol-20 microcomputer, but most of the following discussion applies to any system. The only system-dependent aspect of the interface is generating a compatible strobe signal. The Sol-20 requires a six-microsecond, low-active pulse. The TASA keyboard supplies a continuous, high-active strobe as shown in Fig. 2. While the TASA strobe is ideal for audio feedback and autorepeat circuits, it had to be massaged for my microcomputer system.

The Strobe. The logic for creating a short, low-active strobe pulse from the TASA keyboard's strobe level is shown in Fig. 2b, and represents the only circuitry required to interface

Address correspondence to Douglas H. Haden, PO Box 1296, Ridgecrest, CA 93555.

BIT NUMBERS															
							b_7 →	0	0	0	0	1	1	1	1
							b_6 →	0	0	1	1	0	0	1	1
							b_5 →	0	1	0	1	0	1	0	1
B_G	B_F	B_E	B_D	B_C	B_B	B_A	ROW ↓ / COLUMN →	0	1	2	3	4	5	6	7
			0	0	0	0	0	NUL	DLE	SP	0	@	P	'	p
			0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
			0	0	1	0	2	STX	DC2	"	2	B	R	b	r
			0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
			0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
			0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
			0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
			0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
			1	0	0	0	8	BS	CAN	(	8	H	X	h	x
			1	0	0	1	9	HT	EM	)	9	I	Y	i	y
			1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
			1	0	1	1	11	VT	ESC	+	;	K	[	k	{
			1	1	0	0	12	FF	FS	,	<	L	\	l	|
			1	1	0	1	13	CR	GS	–	=	M	]	m	}
			1	1	1	0	14	SO	RS	.	>	N	∧	n	~
			1	1	1	1	15	SI	US	/	?	O	_	o	DEL

NUL—Null or all zeros
SOH—Start of heading
STX—Start of text
ETX—End of text
EOT—End of transmission
ENQ—Enquiry
ACK—Acknowledge
BEL—Bell or alarm
BS—Backspace
HT—Horizontal tabulation
LF—Line feed
VT—Vertical tabulation
FF—Form feed
CR—Carriage return
SO—Shift out
SI—Shift in
DLE—Data link escape
DC1—Device control 1
DC2—Device control 2
DC3—Device control 3
DC4—Device control 4
NAK—Negative acknowledge
SYN—Synchronous idle
ETB—End of transmission block
CAN—Cancel
EM—End of medium
SUB—Substitute
ESC—Escape
FS—File separator
GS—Group separator
RS—Record separator
US—Unit separator
SP—Space
DEL—Delete

Fig. 1. Key nomenclature and encoded output of the TASA 55 keyboard. The shift and control keys permit the 55-key board to generate all 128 ASCII characters. While the TASA board does not have special-function keys, it is easy to decode selected control characters for cursor control, system reset and the like.

the TASA board to a Sol-20. The circuit is a simple differentiator (also called a half-monostable—see Don Lancaster's *CMOS Cookbook*, pp. 247–249, for more details). I used two 470 pF capacitors in parallel for C_1. Component values are not critical, but may have to be varied for a different microcomputer system. The values of C_1 and R_2 determine the response characteristics of the circuit (more on this below under "Construction"). If an active-high strobe pulse is needed, change the 7404 to a noninverter or use two of the 7404's gates in series.

Auditory Feedback. The lack of either tactile or auditory feedback from keystrokes almost made me desert the TASA keyboard. I'm no great typist at best, but my error rate with the TASA keyboard made it useless. I used two 555 timer circuits to give me a pleasant-sounding bleep on each keystroke. The circuit is shown in Fig. 3. The first 555 is a simple triggered monostable (from Walter Jung's *IC Timer Cookbook*, p. 27) with an output pulse width of $T = 1.1R_2C_3$, or about five ms. The second 555 is a square-wave astable (from pp. 116–118 of Jung's *Cookbook*) with an output frequency variable over the portion of the audio spectrum most attractive for an electronic key click.

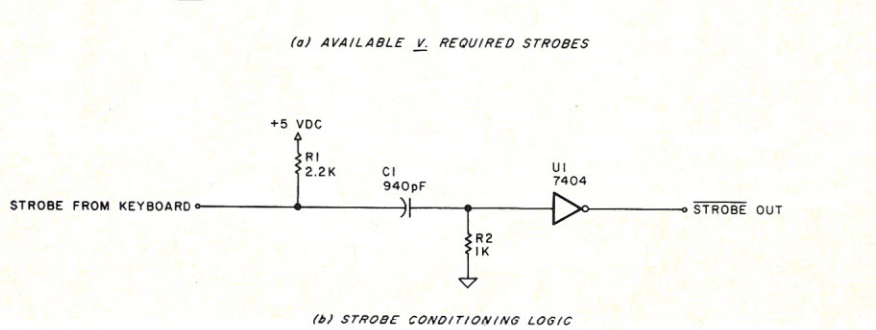


Fig. 2. Strobe timing. For most microcomputers the only circuitry required to interface to the TASA keyboard is a strobe conditioner.

A fixed resistor may be substituted for R_5. The output frequency is $f = 0.722/R_5C_3$. A 47k value for R_5 produces about 1.5 kHz output. By connecting the conditioned strobe to the input of the monostable, and the monostable's output to the input of the astable, you get a five-ms burst of audible tone every time a key is activated. A 2N2222 transistor is used to drive a small speaker, with R_8 as a volume control.

Added Features

My Sol-20 keyboard has discrete keys for several functions which are not available as designated keys on the TASA keyboard. In particular, I need clear screen, repeat, cursor control (left, right, up, down and home) and reset functions. Fortunately, most of these are available on my microcomputer (as on many other systems) via control characters: control-K is a clear screen command, control-A is a left-cursor command and so on.

Unfortunately, repeat and system reset are not available and had to be added in the interface electronics. The easiest to implement was system reset. I used a 7430 eight-input NAND gate and two spare inverters on the 7404 strobe-conditioning chip to detect the character I selected for system reset— a control-delete (the ASCII control character US or binary 00111111). I used a NAND gate because my system requires a low-active system reset.

The last interface circuit, autore-

peat, turned out to be an improvement over the separate-repeat-key feature it replaced. This circuit (purloined from Lancaster's *CMOS Cookbook*, p. 258) waits just over half a second and then, if the input strobe is still active, generates strobe pulses at the rate of five per second. Thus, you need not depress two keys simultaneously; just keeping a key activated will cause it to begin repeating. The autorepeat logic is connected to the rest of the system by C_5. Fig. 3 shows the complete interface circuit diagram.

Construction

Board layout and wiring are no problem at all. Built originally on an AP Products breadboard, the interface was easy to transfer to the general-purpose IC board shown in Photo 2. The microcomputer is connected by a short ribbon cable (the interface board is physically mounted in my Sol-20 during operation) with a nonpolarized female receptacle on the end connecting to the motherboard, and a 14-pin DIP on the interface end, which plugs into the empty socket shown on the board in Photo 2.

The keyboard is connected by a 20-foot ribbon cable—the interface is located in the microcomputer instead of between the keyboard and its stand to keep timing-critical pulses (such as the conditioned strobe) from having to travel over the longer cable. The 20-foot cable lets me take the keyboard from my desk to my easy chair when that is more comfortable. Goal attained: no more hunching over a typewriter or fixed keyboard!

Component values are not especially critical. Just to simplify the board layout, I used a Beckman resistor network for all the pull-up resistors, R_8–R_{15}, and the dual-555 key-click generator was constructed from a single 556 chip. The values of R_2 and C_1 are not critical for my system, as mentioned earlier, but may need to be changed if you are interfacing to a system other than Sol-20. The time relationship $T = 0.8R_2C_1$ can be used to select the values of R_2 and C_1.

A Schmitt trigger can be used in place of the 7404 if it is needed to clean up the shape of the conditioned-strobe pulse. A 50 V dc tantalum was used for C_4 to keep the size small. Since the circuit isn't critical, plain old ceramic disk capacitors were used for C_2 and C_3. (If the monostable false-triggers, connect a small disk capacitor from pin 4 to ground.)

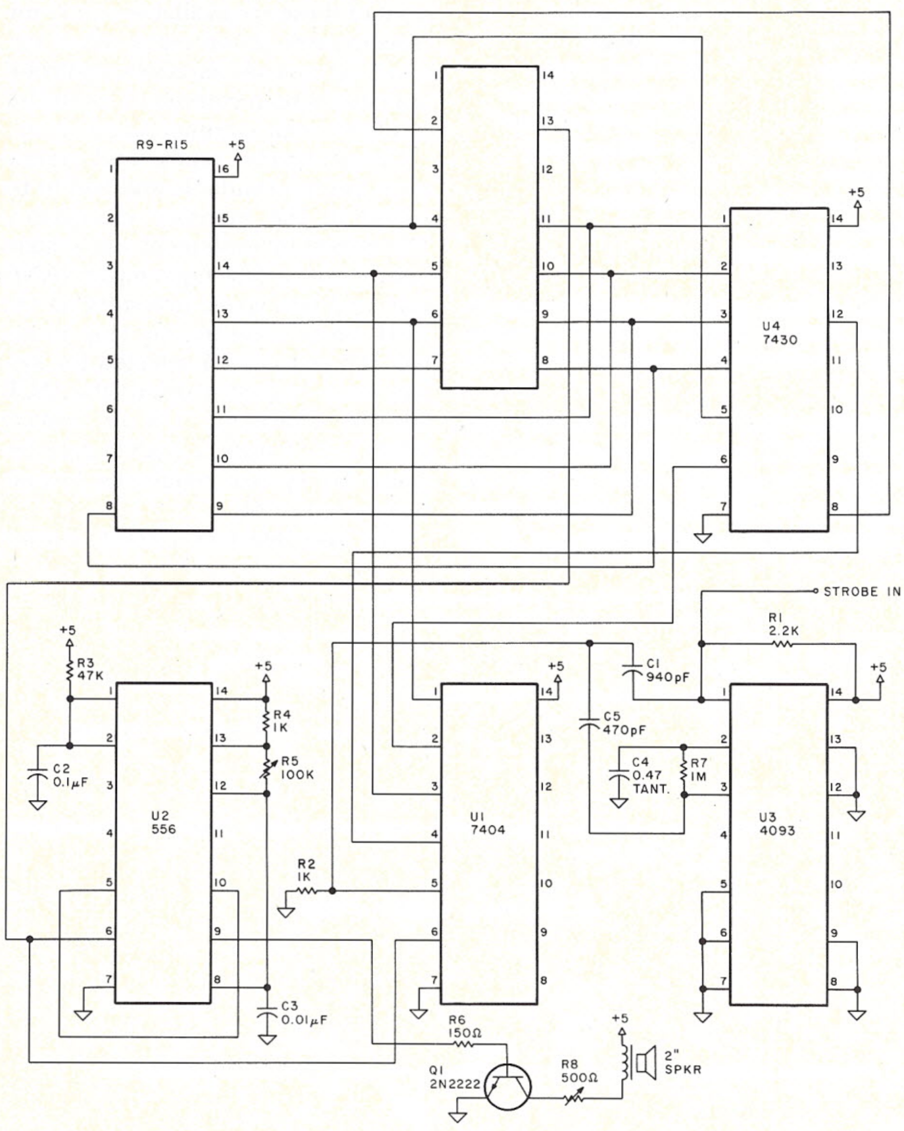


Fig. 3. The complete interface. The strobe is conditioned by a half-monostable (U_1). A dual-555 timer is used to generate an attractive bleep when a key is actuated (U_2); a 4093 CMOS astable generates the auto repeat (U_3); and an eight-wide NAND is used to decode system reset (U_4 and some of U☆). Note that one-half of U_1 *is unused, as is three-fourths of* U_3.

The TASA Board—Pros and Cons

Generally, the TASA keyboard has met my expectations. Designed primarily for use in an "uncontrolled or hostile environment, where operator caution and care cannot be reasonably expected" (a perfect description of my shop/study), the keyboard is rugged and reliable. Its specs include an average expected life of ten years or 50,000 power-on hours (because the keyboard is completely electronic, its life is unaffected by the number of key strokes), operational temperature from −40 °C to +60 °C, shock to 15 g's, and vibration to 1.5 g's (if the keyboard ever gets dirty you can toss it into the dishwasher).

The on-board electronics has its own regulator and needs only 18 V dc, unregulated, at a maximum of 35 mA. All outputs are open-collector type active pull-down logic and need pull-up resistors to drive TTL logic. The specs guarantee only enough power to drive one TTL load, but I have driven several with no problem. Some of the spare 7404 gates in the interface electronics can be used to drive more loads if needed. The outer left and right one-inch edges of the keyboard can be drilled for mounting holes if desired.

For all that, there are aspects of the TASA board that are less than ideal. The connector is a standard 0.156-

NEW! TPM* for TRS-80 Model II
NEW! System/6 Package
Computer Design Labs

Z80* Disk Software

We have acquired the rights to all TDL software (& hardware). TDL software has long had the reputation of being the best in the industry. Computer Design Labs will continue to maintain, evolve and add to this superior line of quality software.
— Carl Galletti and Roger Amidon, owners.

Software with Manual/Manual Alone

All of the software below is available on any of the following media for operation with a Z80 CPU using the CP/M* or similar type disk operating system (such as our own TPM*).

for TRS-80* CP/M (Model I or II)
for 8" CP/M (soft sectored single density)
for 5¼" CP/M (soft sectored single density)
for 5¼" North Star CP/M (single density)
for 5¼" North Star CP/M (double density)

BASIC I
A powerful and fast Z80 Basic interpreter with EDIT, RENUMBER, TRACE, PRINT USING, assembly language subroutine CALL, LOADGO for "chaining", COPY to move text, EXCHANGE, KILL, LINE INPUT, error intercept, sequential file handling in both ASCII and binary formats, and much, much more. It runs in a little over 12 K. An excellent choice for games since the precision was limited to 7 digits in order to make it one of the fastest around. $49.95/$15.

BASIC II
Basic I but with 12 digit precision to make its power available to the business world with only a slight sacrifice in speed. Still runs faster than most other Basics (even those with much less precision). $99.95/$15.

BUSINESS BASIC
The most powerful Basic for business applications. It adds to Basic II with random or sequential disk files in either fixed or variable record lengths, simultaneous access to multiple disk files, PRIVACY command to prohibit user access to source code, global editing, added math functions, and disk file maintenance capability without leaving Basic (list, rename, or delete). $179.95/$25.

ZEDIT
A character oriented text editor with 26 commands and "macro" capability for stringing multiple commands together. Included are a complete array of character move, add, delete, and display function. $49.95./$15.

ZTEL
Z80 Text Editing Language - Not just a text editor. Actually a language which allows you to edit text and also write, save, and recall programs which manipulate text. Commands include conditional branching, subroutine calls, iteration, block move, expression evaluation, and much more. Contains 36 value registers and 10 text registers. Be creative! Manipulate text with commands you write using Ztel. $79.95/$25.

TOP
A Z80 Text Output Processor which will do text formatting for manuals, documents, and other word processing jobs. Works with any text editor. Does justification, page numbering and headings, spacing, centering, and much more! $79.95/$25.

MACRO I
A macro assembler which will generate relocateable or absolute code for the 8080 or Z80 using standard Intel mnemonics plus TDL/Z80 extensions. Functions include 14 conditionals, 16 listing controls, 54 pseudo-ops, 11 arithmetic/logical operations, local and global symbols, chaining files, linking capability with optional linker, and recursive/reiterative macros. This assembler is so powerful you'll think it is doing all the work for you. It actually makes assembly language programming much less of an effort and more creative. $79.95/$20.

MACRO II
Expands upon Macro I's linking capability (which is useful but somewhat limited) thereby being able to take full advantage of the optional Linker. Also a time and date function has been added and the listing capability improved. $99.95/$25.

LINKER
How many times have you written the same subroutine in each new program? Top notch professional programmers compile a library of these subroutines and use a Linker to tie them together at assembly time. Development time is thus drastically reduced and becomes comparable to writing in a high level language but with all the speed of assembly language. So, get the new CDL Linker and start writing programs in a fraction of the time it took before. Linker is compatible with Macro I & II as well as TDL/Xitan assemblers version 2.0 or later. $79.95/$20.

DEBUG I
Many programmers give up on writing in assembly language even though they know their programs would be faster and more powerful. To them assembly language seems difficult to understand and follow, as well as being a nightmare to debug. Well, not with proper tools like Debug I. With Debug I you can easily follow the flow of any Z80 or 8080 program. Trace the program one step at a time or 10 steps or whatever you like. At each step you will be able to see the instruction executed and what it did. If desired, modifications can then be made before continuing. It's all under your control. You can even skip displaying a subroutine call and up to seven breakpoints can be set during execution. Use of Debug I can pay for itself many times over by saving you valuable debugging time. $79.95/$20.

DEBUG II
This is an expanded debugger which has all of the features of Debug I plus many more. You can "trap" (i.e. trace a program until a set of register, flag, and/or memory conditions occur). Also, instructions may be entered and executed immediately. This makes it easy to learn new instructions by examining registers/memory before and after. And a RADIX function allows changing between ASCII, binary, decimal, hex, octal, signed decimal, or split octal. All these features and more add up to give you a very powerful development tool. Both Debug I and II must run on a Z80 but will debug both Z80 and 8080 code. $99.95/$20.

ZAPPLE
A Z80 executive and debug monitor. Capable of search, ASCII put and display, read and write to I/O ports, hex math, breakpoint, execute, move, fill, display, read and write in Intel or binary format tape, and more! on disk

APPLE
8080 version of Zapple

NEW! TPM now available for TRS-80 Model II!

TPM*
A NEW Z80 disk operation system! This is not CP/M*. It's better! You can still run any program which runs with CP/M* but unlike CP/M* this operating system was written specifically for the Z80* and takes full advantage of its extra powerful instruction set. In other words its not warmed over 8080 code! Available for TRS-80* (Model I or II). Tarbell, Xitan DDDC, SD Sales "VERSA-FLOPPY", North Star (SD&DD), and Digital (Micro) Systems. $79.95/$25.

SYSTEM MONITOR BOARD (SMB II)
A complete I/O board for S-100 systems. 2 serial ports, 2 parallel ports, 1200/2400 baud cassette tape interface, sockets for 2K of RAM, 3-2708/2716 EPROM's or ROM, jump on reset circuitry. Bare board $49.95/$20.

ROM FOR SMB II
2KX8 masked ROM of Zapple monitor. Includes source listing $34.95/$15.

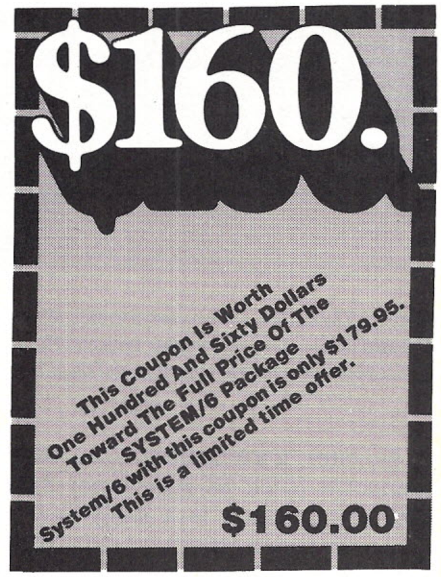
PAYROLL (source code only)
The Osborne package. Requires C Basic 2.
5" disks $124.95 (manual not included)
8" disks $ 99.95 (manual not included)
Manual $20.00

ACCOUNTS PAYABLE/RECEIVABLE (source code only)
By Osborne, Requires C Basic 2
5" disks $124.95 (manual not included)
8" $99.95 (manual not included)
Manual $20.00

GENERAL LEDGER (source code only)
By Osborne. Requires C Basic 2
5" disks $99.95 (manual not included)
8" disks $99.95 (manual not included)
Manual $20.00

C BASIC 2
Required for Osborne software. $99.95/$20.

SYSTEM/6
TPM with utilities, Basic I interpreter, Basic E compiler, Macro I assembler, Debug I debugger, and ZEDIT text editor.
Above purchased separately costs $339.75
Special introductory offer: Only $179.75 with coupon!!



ORDERING INFORMATION
Visa, Master Charge and C.O.D. O.K. To order call or write with the following information.

1. Name of Product (e.g. Macro I)
2. Media (e.g. 8" CP/M)
3. Price and method of payment (e.g. C.O.D.) include credit card info. if applicable.
4. Name, Address and Phone number.
5. For TPM orders only: Indicate if for TRS 80, Tarbell, Xitan DDDC, SD Sales (5¼" or 8"). ICOM (5¼" or 8"), North Star (single or double density) or Digital (Micro) Systems.
6. N.J. residents add 5% sales tax.

Manual cost applicable against price of subsequent software purchase in any item except for the Osborne software.

For information and tech queries call
609-599-2146

For phone orders ONLY call toll free
1-800-327-9191
Ext. 676
(Except Florida)

OEMS
Many CDL products are available for licensing to OEMs. Write to Carl Galletti with your requirements.

* Z80 is a trademark of Zilog
* TRS-80 is a trademark for Radio Shack
* TPM is a trademark of Computer Design Labs. It is not CP/M*
* CP/M is a trademark of Digital Research

Prices and specifications subject to change without notice.

DEALER INQUIRIES INVITED.

COMPUTER DESIGN LABS

✓18

342 Columbus Avenue
Trenton, N.J. 08629

✓See List of Advertisers on page 210

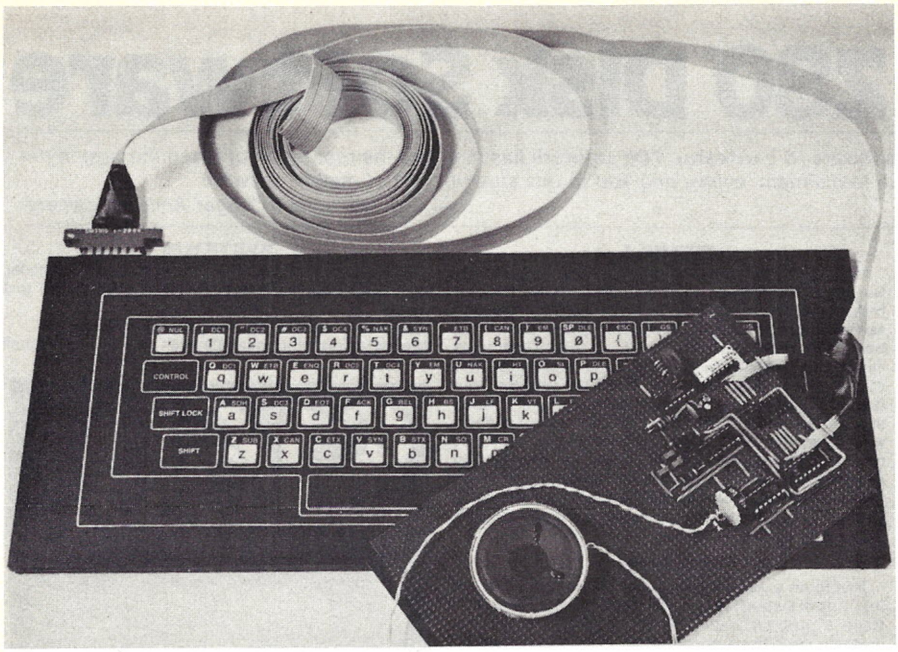


Photo 2. The author's completed interface. The keyboard interface was built on a large general-purpose IC board to provide room for other projects. The 14-pin socket just to the right of the white Beckman resistor network is for the ribbon-cable umbilical to the microcomputer.

inch-center, 2×6-position card edge connector (such as Amphenol 225-20621-101) which is a sadly bulky connector for an otherwise slim, trim component—I have considered soldering the ribbon cable directly to my keyboard. The keyboard can be mounted on a custom stand or its optional TASA stand or, as I prefer, can be used without a stand. If the board is placed directly on bare skin—say, on your bare legs—it will sometimes malfunction, but a layer or two of clothing will prevent this problem. Perhaps the only real shortcoming of the keyboard is the lack of special-function keys. I would rather have had specific cursor-control keys than the parity (bit b_7) and parity-select (odd/even) features.

Conclusion

The interface circuits described above would work with almost any keyboard-to-microcomputer connection. They have given my TASA keyboard enough power to be the only front panel I use. I recommend the bleeper for any nondepressible-key keyboard. For systems requiring more special-function keys (for cursor control, for example), the logic used here to detect a control-delete for system reset could be repeated. For systems with no +18 V dc conveniently available, the low (35 mA) current requirement and on-board regulator make it easy to construct either a small, unregulated stand-alone +18 V dc supply, or a small booster supply to place in series with a lower voltage supply.

Since I have little need for a caps-lock feature the interface doesn't include one. The articles written by Terry Conboy and Michael Dunlavey, listed in my references, describe simple circuits for caps-lock of ASCII keyboards. They could be activated/deactivated by control characters decoded like system reset in this interface.■

References

Chamberlin, Hal. "Keyboards," *Popular Electronics*, March 1980, pp. 84–87.

Conboy, Terry. "Alpha Lock for Your ASCII Keyboard," *Byte*, January 1980, pp. 156–158.

Dunlavey, Michael R. "Caps Lock, Not Shift Lock," *Kilobaud Microcomputing*, October 1979, p. 134.

Jung, Walter G. *IC Timer Cookbook*. Howard W. Sams: Indianapolis, 1977.

Lancaster, Don. *CMOS Cookbook*. Howard W. Sams: Indianapolis, 1977.

Mangieri, Adolph A. "Extra Keyboards for Microcomputers," *Popular Electronics*, April 1980, pp. 54–59.

✓See List of Advertisers on page 210

You'll know the contents of any memory location—in binary, hex, decimal and ASCII—with this useful memory display program. In North Star BASIC.

Here's Where It's At

By Robert J. Roby

```
   LOCATION                              CONTENTS
DECIMAL    HEX          BINARY       HEX      DECIMAL        ASCII
4          0004       1100  0011     C3         195
8220       201C       1100  0011     C3         195
8221       201D       0010  0110     26          38            &
38         0026       1010  0001     A1         161
1000       03E8       0111  1110     7E         126            ~
1001       03E9       0100  0101     45          69            E
1002       03EA       0111  1000     78         120            x
1003       03EB       0110  0001     61          97            a
1004       03EC       0110  1101     6D         109            m
1005       03ED       0111  0000     70         112            p
1006       03EE       0110  1100     6C         108            l
1007       03EF       0110  0101     65         101            e
1008       03F0       0010  0000     20          32
1009       03F1       0010  0001     21          33            !
 Q
READY
```

Sample run.

B	decimal byte address
B1	decimal contents of byte
D	temporary storage for decimal address
D1	temporary storage for decimal to hex conversion
I	loop index
J	loop index
L2	number of significant figures in hex address string
P	bit pointer for decimal to binary conversion
R	remainder in decimal to hex conversion
Z	top memory address +1, used in overrange calculation
A$	string storage for binary contents
B$	input string for decimal address
H$	string storage for hex output
H1$	string used in decimal to hex conversion
L1$	temporary string storage for hex address

Variable usage.

Often when working with BASIC, I want to examine the contents of a memory location. But due to the base 10 tyranny of BASIC, the PEEK and EXAM functions only display the data in decimal form. Machine-language monitors usually only show the data in hexadecimal, or perhaps ASCII (American Standard Code for Information Interchange, the character codes used by most terminals). And nothing ever shows the actual binary representation.

These displays are adequate if you have a conversion table handy, but if you are as lazy as I am you probably think the computer should do this kind of work for you. MEMLIST, written in North Star BASIC, might be the answer.

Once the program is running and the heading is printed, just type in the decimal memory address you want to display, followed by a carriage return. The computer then fills in the other five columns: the hex representation of the location and the contents in binary, hex, decimal and ASCII (see Sample run).

Robert J. Roby, 1209 Driftwood Drive, Euless, TX 76039.

```
100 REM   MEMLIST    25 MAY 1980
110 REM   THIS PROGRAM ACCEPTS DECIMAL MEMORY ADDRESSES
120 REM   AS INPUT AND PRINTS THE CONTENTS IN
130 REM   DECIMAL, HEX, BINARY AND ASCII.
140 REM   THE PROGRAM IS WRITTEN IN NORTHSTAR BASIC.
150 REM   FNB RETURNS THE VALUE OF THE PTH BIT OF
160 REM   THE BYTE AT LOCATION B (1 OR 0).
170 DEF FNB(B,P)=INT(B/2^P)/2<>INT(INT(B/2^P)/2)
180 DIM H1$(16)\H1$="0123456789ABCDEF"\Z=65536
190 REM SET UP OUTPUT HEADING
200 ! TAB(8),"LOCATION",TAB(40),"CONTENTS"
210 ! TAB(4),"DECIMAL",TAB(14),"HEX",TAB(27),"BINARY",
220 ! TAB(38),"HEX",TAB(46),"DECIMAL",TAB(58),"ASCII"
230 INPUT1 "     ",B$\REM GET ADDRESS IN STRING FORM
240 IF B$="Q" THEN END
250 IF  B$="" THEN GOSUB 400 ELSE B= VAL(B$)
260 REM CORRECT FOR RANGE
270 FOR J=B TO Z STEP -Z\B=B-Z\NEXT
280 REM BUILD STRING CONTAINING BINARY EQUIVALENT OF BYTE
290 B1=EXAM(B)
300 FOR P=0 TO 7\A$(8-P,9-P)=STR$(FNB(B1,P))\NEXT
310 REM ROUTINE TO GET HEX ADDRESS AND HEX CONTENTS
320 D=B\GOSUB 410\L1$=H$\GOSUB 490\D=B1\GOSUB 410
330 REM PRINT OUTPUT LINE
340 ! TAB(14),L1$,TAB(25),A$(2,5),"  ",A$(6,9),TAB(39),
350 ! H$,TAB(48),%3I,B1,TAB(60),
360 REM PRINT ASCII CONTENTS IF IT IS PRINTABLE CHARACTER
370 IF B1>32 AND B1<127 THEN ! CHR$(B1) ELSE !
380 GOTO 230\REM GO GET NEXT ADDRESS
390 REM ROUTINE TO HANDLE AUTO INCREMENT THROUGH MEMORY
400 B=B+1\B$=STR$(B)\! B$(2),\RETURN
410 REM H1$ MUST BE DIMENSIONED AND SET TO "0123456789ABCDEF"
420 REM BEFORE CALLING THIS DECIMAL TO HEX CONVERSION ROUTINE
430 H$=""
440 D1=INT(D/16)\R=D-D1*16\GOSUB 470
450 IF D1<16 THEN 460 ELSE D=D1\GOTO 440
460 R=D1
470 H$=H1$(R+1,R+1)+H$\RETURN
480 REM PAD HEX VALUE TO 4 CHARACTERS WITH LEADING ZEROS
490 L2=LEN(L1$)\FOR I=L2 TO 3\L1$="0"+L1$\NEXT\RETURN
```

Program listing.

It is often convenient to look at a consecutive series of locations. To do this, type in the first address of the series and wait for it to be displayed. For each consecutive address simply type a carriage return. MEMLIST will automatically increment the address and display the contents. When you are through looking at memory, type a Q to exit the program.

The program output is formatted to fit on a 64-character screen.■

Two convenient additions to the Exidy Sorcerer's capabilities: line renumbering and merging.

Teach a Sorcerer New Tricks

By C. Kevin McCabe

The Exidy Sorcerer has a fairly extensive bag of tricks, but does lack a few features found in other BASIC interpreters. Two obvious shortcomings are the lack of a line renumbering capability and the inability to merge two or more taped programs into a single routine.

Although these abilities aren't part of the standard interpreter, they *can* be implemented. All it takes is an extra wave of the Sorcerer's wand, a few magic programming words and some knowledge of how BASIC programs are stored in memory.

Sorcerer's Microsoft 8K BASIC is not too different from the TRS-80 or PET versions. All program lines are stored in memory as ASCII character strings, with the lowest-numbered lines at the lowest memory addresses. Within the text of each stored program line, certain keywords—IF, FOR, GOTO and all the other commands and symbols in BASIC—are not spelled out. Instead, each such keyword is represented by a single-byte character, or token, having an unsigned integer value between 129 and 198 (80H and C6H).

The text of each full program line, which may include several statements separated by colons, is preceded by a four-byte header (Fig. 1). These four bytes specify the number of the current line, and a 16-bit address pointer to the beginning of the next full line's header. Both of these values are given in the form of a two-byte integer binary number. It's easy to switch between integer binary and decimal values of line numbers or addresses:

Decimal value = low-order byte + (256 * high-order byte)
High-order byte = INT (decimal value / 256)
Low-order byte = decimal value − (256 * high-order byte)

Like most 16-bit values in memory, the low-order byte is stored in the first (lowest address) location, followed by the high-order byte. For example, if the header of the *next* full line starts at (decimal) address 500, the first two bytes of the *current* header would contain (decimal) 244 and 1. If the last two header bytes contain (decimal) 232 and 3, the number of the *current* line must be 1000.

At the end of each full program line stored in memory, following the final ASCII character, is a single byte with a zero value. BASIC uses this byte to identify the end of program lines while running or listing the program. The header for the next full line begins at the address immediately after the zero-valued byte, which—barring glitches—corresponds to the address pointer in the preceding header. This system of address pointers creates a "linked list," such as used for data base operations, although the "data" here is really program lines.

All BASIC programs begin at the same location in memory; the first byte of the first header is always located at decimal address 469 (01D5H). The end of the program is shown by an address pointer of 0000 in the final header. No program text follows this special pointer.

RENUM

Now, let's use this information to create RENUM, a renumbering routine. It could be done in machine language—but writing machine language routines is not exactly the most thrilling way to spend an afternoon. In keeping with the KISS principle ("Keep It Simple, Stupid"), the renumbering routine can be written in BASIC. Sure, it will be slower—and as will be shown it will have some minor shortcomings—but a BASIC routine is certainly easier to write and debug!

Listing 1 shows the result. Both RENUM and the BASIC program to be renumbered must be in memory at the same time, as a single program (more about this later). A direct mode command of RUN 65000 begins execution. Since the routine being renumbered isn't being run, there's no problem if a variable name is duplicated in RENUM. If the program to be renumbered already has line numbers greater than or equal to 65000, though, you're out of luck.

Lines 65015–45 of RENUM ask for the lowest line number to be changed, that line's new number (which forms a "base" for the rest of the routine) and the increment to be added to

Address correspondence to C. Kevin McCabe, 115 S. LaSalle St., Chicago, IL 60603.

each following line number. Lines 65050–85 skip through the entire program, using the information in one header to find the next and count the number of lines to be renumbered.

A renumbering routine must do more than change the line numbers in the headers, though. The program text also contains line numbers associated with GOSUB, GOTO and THEN statements. Once the values in the headers are changed, the RENUM must still be able to relate the new line numbers to the corresponding old numbers. The array LNS is used as a list of the original (decimal) line numbers that are to be changed by the routine. Lines 65090–130 fill the array and change the line number values found in the header areas.

The new line numbers are calculated simply, as the sum of the new "base" number plus the appropriate multiple of the increment between line numbers. It's no coincidence that the "appropriate multiple" for the increment is the same as the array location of the old line number. For example, assume line number 1459 is the fourth line in a program, and that the first program line's number is to be changed from 13 to 100, with an increment of 50 between lines. Line 1459 will be changed to 250 (100 + 3 * 50), and LNS(3) will contain the value 1459.

Now things become a bit more complicated—but still easier than machine language. Each program line may contain zero, one or more GOTO, GOSUB or THEN statements. The first two will always be followed by at least one line number reference, and by multiple references separated by commas in the case of ON-GOTO or ON-GOSUB statements. A THEN statement can be followed by a line number reference, a GOTO or GOSUB statement with a line number reference, or a non-referenced statement such as PRINT or STOP.

Remember that keywords such as GOTO or THEN aren't spelled out in memory; they use special single-byte tokens. A byte having a decimal value of 137 is a token for GOTO. The GOSUB token is a value of 141, and THEN is 162. Lines 65135–75 locate the zero-valued byte that immediately precedes a line's header, as well as the zero-valued byte that ends the line, and search for these three tokens within the line of program text.

When a token is found, lines 65180–200 check to see if the next non-blank

```
65000 REM--Function and variable initialization
65005 : DEF FNDEC(J) = PEEK(J) + 256 * PEEK(J+1)
65010 : CNT = 0: ZB = 468: LOWLN = FNDEC(ZB+3)
65015 REM--Operator input
65020 : INPUT "First change, change to, increment";FC,NLN,DELTA
65025 : IF FC <= NLN OR FC = LOWLN THEN 65045
65030 : PRINT "No, first value input must be less than second,"
65035 : PRINT "  if only part of program is to be renumbered."
65040 : PRINT "Try again, please!": PRINT: GOTO 65020
65045 : IF DELTA <= 0 THEN PRINT "Bad increment--";: GOTO 65040
65050 REM--Count number of lines to change, & dimension array
65055 : NZB = FNDEC(ZB+1) - 1: CURLN = FNDEC(ZB+3)
65060 : IF CURLN = 65000 THEN 65075
65065 : IF CURLN >= FC THEN CNT = CNT + 1
65070 : ZB = NZB: HILN = CURLN: GOTO 65055
65075 : DIM LNS(CNT-1): PRINT: PRINT "Lines"; FC; "thru"; HILN;
65080 : PRINT "become lines"; NLN; "thru"; NLN + DELTA*(CNT - 1)
65085 : PRINT "A total of"; CNT; "line numbers changed."
65090 REM--Change line numbers in headers, & fill array
65095 : ZB = 468: J = 0
65100 : NZB = FNDEC(ZB+1) - 1: CURLN = FNDEC(ZB+3)
65105 : IF CURLN < FC THEN ZB = NZB: GOTO 65100
65110 : LNS(J) = CURLN: NWLN = NLN + J*DELTA: J = J + 1
65115 : POKE ZB+3, NWLN - 256*INT(NWLN/256)
65120 : POKE ZB+4, INT(NWLN/256): ZB = NZB
65125 : IF J < CNT THEN 65100
65130 : PRINT: PRINT "Header changes complete.": PRINT
65135 REM--Seek out "GOTO", "GOSUB", and "THEN" tokens
65140 : ZB = 468
65145 : ADR = ZB + 4: NZB = FNDEC(ZB+1) - 1
65150 : CURLN = FNDEC(ZB+3): IF CURLN < 65000 THEN 65160
65155 : PRINT: PRINT: PRINT "Routine completed.": END
65160 : PRINT: PRINT CURLN;
65165 : ADR = ADR + 1: CV = PEEK(ADR)
65170 : IF ADR >= NZB THEN ZB = NZB: GOTO 65145
65175 : IF CV <> 137 AND CV <> 141 AND CV <> 162 THEN 65165
65180 REM--Find line number reference, if any, after token
65185 : CMDADR = ADR: N$ = ""
65190 : ADR = ADR + 1: CV = PEEK(ADR): IF CV = 32 THEN 65190
65195 : IF CV = 137 OR CV = 141 THEN 65185
65200 : IF CV < 48 OR CV > 57 THEN 65165
65205 REM--Replace old line reference with new, if possible
65210 : N$ = N$ + RIGHT$(STR$(CV - 48),1)
65215 : ADR = ADR + 1: CV = PEEK(ADR)
65220 : IF CV >= 48 AND CV <= 57 THEN 65210
65225 : GOSUB 65235: IF CV = 44 THEN 65185
65230 : GOTO 65170
65235 REM--Subroutine--replace line reference with new number
65240 : N = VAL(N$): IF N < FC THEN RETURN
65245 : FOR J = 0 TO CNT - 1: IF LNS(J) = N THEN 65255
65250 : NEXT J: PRINT "Line #"; N; "is undefined.": RETURN
65255 : NWLN = NLN + J*DELTA: SPACE = ADR - CMDADR - 1
65260 : N$ = STR$(NWLN): N$ = RIGHT$(N$,LEN(N$) - 1)
65265 : IF LEN(N$) < SPACE THEN N$ = " " + N$: GOTO 65265
65270 : IF LEN(N$) = SPACE THEN 65285
65275 : PRINT "NOTE--operator must change"; N; "to "; N$;
65280 : PRINT " in this line.": RETURN
65285 : FOR J = 1 TO SPACE: CV$ = MID$(N$,J,1)
65290 : CX = VAL(CV$) + 48: IF CV$ = " " THEN CX = 32
65295 : POKE CMDADR + J, CX: NEXT J: RETURN
```

Listing 1. The RENUM renumbering routine.

characters are a line number reference or another token (to cover the case of THEN GOTO or THEN GOSUB). If not, the search for tokens resumes.

Lines 65205–30 pull out the original reference following a token, digit-by-digit, and convert it to a string. The subroutine at lines 65235–95 will check the LNS array for the value found, and—most of the time—calculate the corresponding new value and poke it in memory following the token. The search then resumes for additional tokens, or other line number references separated by commas.

Most of the time? Well, yes—there are two instances where the original value will not be changed. The first instance is when the reference is to an undefined line. If the original program statement is "GOTO 999" but no line 999 existed in the first place, there's obviously no way for RENUM to calculate a corresponding new line number. An advisory message is printed whenever such an undefined line reference is located.

The other instance is less obvious. Suppose the program statement "GOSUB 10" is to be renumbered, where the original line number 10 has been renumbered to line 1000. In memory, "GOSUB 10" takes up four bytes—

for a token, a blank and the two digits—but "GOSUB1000" requires five bytes, even without a blank space between the token and the line reference. There's not enough room to insert the new number after the token.

The most obvious solution would be to shift the entire program one or two bytes forward in memory, but it won't work. The headers, remember, contain link pointers—addresses for following lines—which would be incorrect after such a shift. Adding an increment to each pointer is no good either, as that would require the renumbering routine itself to be changed while it is running.

In keeping with the KISS principle, RENUM takes the simplest approach. Whenever a new reference number won't fit the available space, the old number is left unchanged and a warning message is sent to the screen. The message specifies the new number of the affected line and both the old and new line references. At the completion of RENUM, simply key in a corrected line. A little care in writing original programs, with an extra space or two before line references, will do away with the need for such changes most of the time.

Other Shortcomings

Remember that both the renumbering routine and the program to be renumbered must be in memory at one time, combined as a single program. Using the renumberer as a development tool is simple enough; just load RENUM from tape first, then write the program as usual with line numbers below 65000. The program under development can be renumbered as many times as needed until completed; then the lines of RENUM can be deleted.

But what about existing programs? If a program to be renumbered is already in memory, a CLOAD RENUM will wipe it out. If RENUM is already in memory, loading the other program will wipe it out. That means

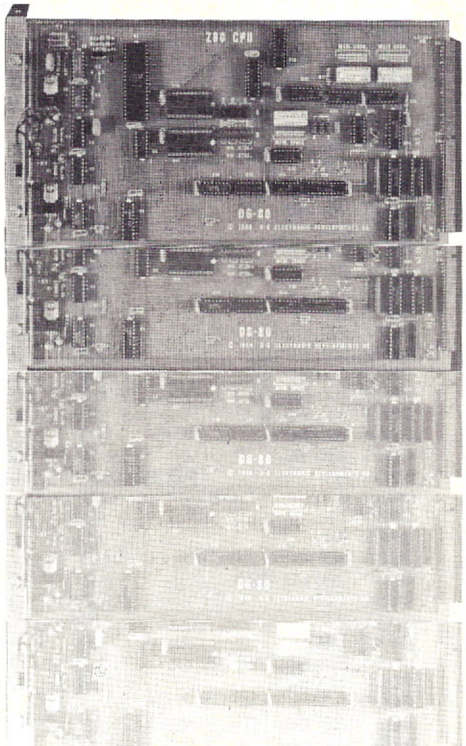
Listing 2. MERGE—machine-language routine to combine BASIC programs.

```
0000                    ; MERGE
0000                    ; BASIC PROGRAM COMBINER FOR THE SORCERER
0000                    ; C. KEVIN MCCABE
0000                    ;
0000                    TOBAS   EQU   00C0
0000                    NBM     EQU   00C2
0000                    CURND   EQU   00C4
0000                    BOBAS   EQU   00C6
0000                    ACP     EQU   00C8
0000                    OFST    EQU   00CA
0000                    ERROR   EQU   00D0
0000                    END1    EQU   00E0
0000                    END2    EQU   00F0
0000                    WARM    EQU   0E003
0000                    SEND    EQU   0E00C
0000                    ; PART ONE--LOAD BASIC PROGRAM #1, THEN RUN
0000                    ;             THIS PART WITH MONITOR GO 0000
0000 2A 00 F0                   LD    HL,(0F000)   ; SET UP STACK
0003 2E 90                      LD    L,90
0005 F9                         LD    SP,HL
0006 01 91 01                   LD    BC,0191      ; STACK SPACE
0009 B7                         OR    A
000A ED 42                      SBC   HL,BL        ; CALC. TOP OF
000C 22 C0 00                   LD    (TOBAS),HL   ;  FREE MEMORY
000F CD AA 00                   CALL  LBYTE        ; FIND LAST BYTE OF
0012                                               ;  BASIC PROGRAM #1
0012 01 D4 01                   LD    BC,01D4      ; CALC. # OF BYTES
0015 B7                         OR    A            ;  TO SHIFT
0016 ED 42                      SBC   HL,BC
0018 22 C2 00                   LD    (NBM),HL
001B ED 4B C2 00                LD    BC,(NBM)     ; SHIFT PROGRAM #1
001F ED 5B C0 00                LD    DE,(TOBAS)   ;  TO UPPER MEMORY
0023 2A C4 00                   LD    HL,(CURND)
0026 ED B8                      LDDR
0028 13                         INC   DE
0029 ED 53 C6 00                LD    (BOBAS),DE
002D 21 E0 00                   LD    HL,END1      ; SCREEN MESSAGE,
0030 06 0F                      LD    B,0F         ;  THEN TO MONITOR
0032 18 6A                      JR    NCHAR-$
0034 00                         NOP
0035                    ; PART TWO--LOAD BASIC PROGRAM #2, THEN RUN
0035                    ;             THIS PART WITH MONITOR GO 0035
0035 CD AA 00                   CALL  LBYTE        ; CALC. LAST BYTE OF
0038                                               ;  BASIC PROGRAM #2
0038 B7                         OR    A            ; SUFFICIENT MEMORY?
0039 2A C6 00                   LD    HL,(BOBAS)
003C ED 4B C4 00                LD    BC,(CURND)
0040 ED 42                      SBC   HL,BC
```

More

TOMORROW'S answers for the H8® TODAY!

DG Electronic Developments provides the only enhancements to put the Heath® H8 ahead ***TODAY*** while maintaining its high standards, broad flexibility and high reliability.



DG-80 (Z80® Based CPU)

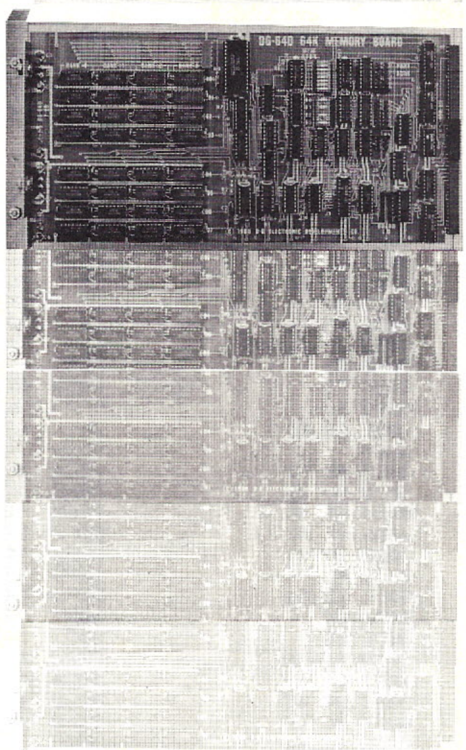
The DG-80 offers the **BEST** foundation for the H8 to build on for the future. Compare the DG-80 to your current Heath® 8080 CPU and see the advantages of the DG-80's superior speed and flexibility. The Z80 gives twice the processing speed coupled with twice the instruction set making your H8 a truly professional microsystem. The DG-80 aids in efficient writing of utility programs, user routines, sub-routines and use of assemblers.

DG-80 $249 (Documentation only $25.)

DG-FP8/ADP4 (DG80 Front Panel Monitor)

The DG-FP8/ADP4 firmware/hardware package brings the world of standardization to your H8. The FP8/ADP4 allows the full use of currently available "STANDARD" CPM software, with 4 MHz operation. You can still use your HDOS and all its related software, by simply putting a disk in your drive. The DG firmware package gives you use of the full 64K of RAM available on the 64D or 64D5. The FP8/ADP4 firmware also offers many extras including hexadecimal OR split octal display; simplification of register and memory contents display; and the full use of all Z80 registers. Supports 5¼ and 8" Heath disk drives.

DG-FP8 $69.95. DG-ADP4 $19.95. Purchased together $79.95.
(Documentation only — source code not included $15.)



DG-64D (64K RAM Memory Board)

The DG-64D is **THE BEST** memory board available for the H8! Along with the DG-80 CPU, the 64D gives a full 64K of addressable memory. Its built-in flexibility ensures future compatibility. Multi-user configurations can be achieved accessing up to eight 64D's on line with the hardware/software bank select features. Memory contents are maintained and protected during extended wait states by asynchronous refresh.

DG-64D priced from $333 (ØK) to $529 (64K). (Documentation only $15.)

DG-64D5 (5-Volt Power Supply 64K RAM)

The DG-64D5 memory board offers the **ultimate** in low power consumption for the H8 computer! The 64D5 is primarily designed for use with the Heath Color Graphics board. In addition to offering **ALL** of the features of the standard 64D, the 64D5 utilizes reliable **single-supply** 5-volt-only dynamic RAMs.

Priced from $333 (ØK) to $799 (64K). (Documentation only $15.)

CP/M (Standard V2.2 Disk Operating System)

The final component for the **"STANDARD"** H8 system is Digital Research's CP/M® V2.2. Our full line of H8 enhancements in combination with CP/M gives the user unparalleled flexibility and software availability.

CPM V2.2 $130. (Documentation only $25.)

All products supplied with complete documentation.

Please call or write for additional information.

AVAILABLE AT MANY HEATHKIT ELECTRONIC CENTERS

Heath® and H8® are registered trademarks of Heath Corporation, Benton Harbor, Michigan. CP/M® is the registered trademark of Digital Research of Pacific Grove, California. Z80® is the registered trademark of Zilog Corporation.

D·G ELECTRONIC DEVELOPMENTS CO.

145

Ordering Information: Products listed available from DG Electronic Developments Co., P.O. Box 1124, 1827 South Armstrong, Denison, Tx. 75020. Check, Money Order, VISA or MasterCard accepted. Phone orders (charge only) call (214) 465-7805. No COD's. Freight prepaid. Allow 3 weeks for personal checks to clear. Texas residents add 5%. Foreign orders add 30%. Prices subject to change without notice.

Listing 2 continued.

```
0042 38 55            JR    C,OVFLO-$      ; IF NOT, ABORT RUN
0044 2A C4 00         LD    HL,(CURND)     ; SET ADDRESS POINTER
0047 2B               DEC   HL             ;  TO START OF FINAL
0048 22 C8 00         LD    (ACP),HL       ;  HEADER OF PROG #2
004B ED 4B C2 00      LD    BC,(NBM)       ; SHIFT PROGRAM #1 TO
004F 2A C6 00         LD    HL,(BOBAS)     ;  DIRECTLY FOLLOW
0052 ED 5B C8 00      LD    DE,(ACP)       ;  PROGRAM #2
0056 ED B0            LDIR
0058 2A C8 00         LD    HL,(ACP)       ; FIND OFFSET TO ADD
005B 01 D5 01         LD    BC,01D5        ;  TO LINK POINTERS
005E B7               OR    A              ;  OF PROGRAM #1
005F ED 42            SBC   HL,BC
0061 22 CA 00         LD    (OFST),HL
0064 ED 5B CA 00      LD    DE,(OFST)
0068 2A C8 00   LOOP2 LD    HL,(ACP)       ; GET VALUE OF LINK
006B 4E               LD    C,(HL)         ;  POINTER FROM
006C 23               INC   HL             ;  CURRENT HEADER
006D 46               LD    B,(HL)
006E 3E 00            LD    A,0            ; IS THIS THE FINAL
0070 B8               CP    B              ;  0000 LINK POINTER?
0071 20 15            JR    NZ,PRUPD-$     ; NON-ZERO, SO JUMP
0073 B9               CP    C
0074 20 12            JR    NZ,PRUPD-$     ; NON-ZERO, SO JUMP
0076 23               INC   HL             ; ZERO LINK FOUND,
0077 23               INC   HL             ;  SO UPDATE POINTERS
0078 22 B7 01         LD    (01B7),HL      ;  IN BASIC CONTROL
007B 22 B9 01         LD    (01B9),HL      ;  AREA
007E 22 BB 01         LD    (01BB),HL
0081 21 F0 00         LD    HL,END2        ; SCREEN MESSAGE,
0084 06 04            LD    B,04           ;  THEN TO MONITOR
0086 18 16            JR    NCHAR-$
0088 60         PRUPD LD    H,B            ; UPDATE LINK POINTER
0089 69               LD    L,C            ;  VALUE BY ADDING
008A 19               ADD   HL,DE          ;  OFFSET VALUE
008B 44               LD    B,H
008C 4D               LD    C,L
008D 2A C8 00         LD    HL,(ACP)
0090 71               LD    (HL),C
0091 23               INC   HL
0092 70               LD    (HL),B
0093 ED 43 C8 00      LD    (ACP),BC       ; UPDATE ADDRESS OF
0097 18 CE            JR    LOOP2-$        ;  CURRENT HEADER
0099                                       ;  AND REPEAT LOOP
0099 21 D0 00   OVFLO LD    HL,ERROR       ; SEND OVERFLOW
009C 06 0E            LD    B,0E           ;  MESSAGE
009E 7E         NCHAR LD    A,(HL)         ; SEND TO OUTPUT
009F CD 0C E0         CALL  SEND
00A2 23               INC   HL
00A3 05               DEC   B
00A4 20 F7            JR    NZ,NCHAR-$
00A6 C3 03 E0         JP    WARM           ; JUMP TO MONITOR
00A9 00               NOP
00AA 21 D5 01   LBYTE LD    HL,01D5        ; SUBROUTINE TO GET
00AD 4E         LOOP1 LD    C,(HL)         ;  ADDRESS OF LAST
00AE 23               INC   HL             ;  BYTE OF A BASIC
00AF 46               LD    B,(HL)         ;  PROGRAM'S FINAL
00B0 3E 00            LD    A,0            ;  0000 LINK POINTER
00B2 B8               CP    B
00B3 20 07            JR    NZ,UPDAT-$
00B5 B9               CP    C
00B6 20 04            JR    NZ,UPDAT-$
00B8 22 C4 00         LD    (CURND),HL
00BB C9               RET
00BC C5         UPDAT PUSH  BC
00BD E1               POP   HL
00BE 18 ED            JR    LOOP1-$
00C0                  DEFS  10             ; VARIABLES AREA
00D0 4F 56            DEFW  564F           ;'OVERFLOW ABORT'
00D2 45 52            DEFW  5245
00D4 46 4C            DEFW  4C46
00D6 4F 57            DEFW  574F
00D8 20 41            DEFW  4120
00DA 42 4F            DEFW  4F42
00DC 52 54            DEFW  5452
00DE 00               NOP
00DF 00               NOP
00E0 50 50            DEFW  5050           ;'PP + CLOAD NEXT'
00E2 20 2B            DEFW  2B20
00E4 20 43            DEFW  4320
00E6 4C 4F            DEFW  4F4C
00E8 41 44            DEFW  4441
00EA 20 4E            DEFW  4E20
00EC 45 58            DEFW  5845
00EE 54               DEFB  54
00EF 00               NOP
00F0 44 4F            DEFW  4F44           ;'DONE'
00F2 4E 45            DEFW  454E
```

that the renumbering routine must be keyed in, line-by-line and character-by-character, each time it's needed.

Or does it? There *is* a better way, a way to load two or more BASIC programs from tape and combine them into a single, merged routine.

MERGE

MERGE (Listing 2) is a two-part machine-language routine to combine BASIC programs. MERGE may be used to add RENUM to an existing program or used to combine common library utility operations with an application program.

Remember that BASIC program lines are stored as strings of ASCII characters. The first four bytes of each line are a header, containing the line number of the current line, and a link pointer specifying the address of the next full line's header. The final header has a link pointer value of 0000, denoting the end of the program. A single byte, with a value of zero, separates the end of each line and the header of the following line (see Fig. 1). All BASIC programs begin at address 01D5H (469 decimal). The address of the final 0000 link pointer can be found by skipping through the program, header-by-header, and checking for a zero-value link pointer.

Fig. 2 shows the Sorcerer's memory allocations for a BASIC program. The address TIMEM denotes the top of installed memory (other than the ROM PAC space and video RAM, from C000H to FFFFH). For a 16K machine, TIMEM is 3FFFH; for 32K, 7FFFH is the top of user memory. The upper 111 bytes are reserved for the monitor's control memory. The monitor stack grows downward from below the control area.

Below the stack, BASIC dynamically allocates space for string and numeric variables and arrays, along with another stack and some free RAM. These areas "float" as the BASIC program is expanded or contracted, and also change during execution of the program. To keep track of the different areas, BASIC has its own program control memory at addresses 0100H through 01D4H. Address pointers to the start of the numeric variable storage area and the first and last bytes of the array area are placed in the control area, beginning at 01B7H, 01B9H and 01BBH, respectively.

BASIC expects to find these values in place prior to running or listing the

program; if not, the machine goes off on a tangent, requiring a reset and loss of the program in memory. The interpreter takes care of the task during normal additions or deletions from programs and after loading a program from tape.

Combining Programs

First, a BASIC program is keyed or loaded into memory in the usual way. Of the two or more programs to be merged, the first one *must* have the highest line numbers. The line numbers appearing in each program must be unique, and each must be greater than the highest line number in later programs to be added. The RENUM program starts with line 65000, so it can be used with virtually any other program to adjust line numbers as necessary.

Once the first BASIC program—let's call it HINUM—is in memory, the BYE command is used to exit to the monitor. The MERGE machine-language routine is loaded in, occupying the free space at addresses 0000H through 00FFH. The monitor command GO 0000 begins the first phase of MERGE.

MERGE determines the highest memory location that can be used, allowing space for both the monitor stack and its control memory. To make the job easier, MERGE uses the helpful fact that the two-byte address of TIMEM is stored at F000H by the monitor.

The length of the program is determined much as in RENUM, by skipping through the program header-by-header and looking for a link pointer of 0000H. Once the size of the program has been found, the entire HINUM program is shifted as a block into a location higher in memory, just below the monitor stack area. BASIC can't do this, but a machine-language routine like MERGE can treat the stored program as just another bunch of bytes to be manipulated. At the end of the shift operation, a screen message tells the operator to exit the monitor and load the next BASIC program.

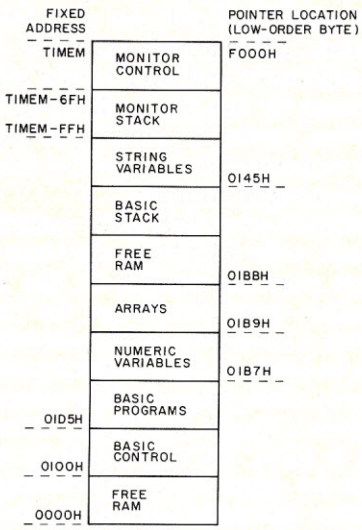


Fig. 1. BASIC program storage format.

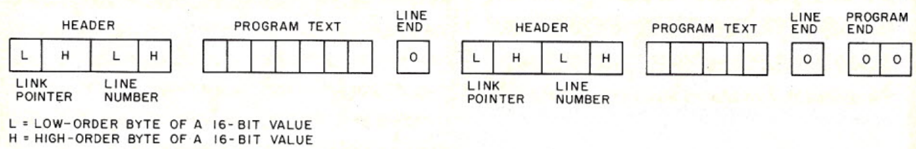


Fig. 2. BASIC memory map.

✓See List of Advertisers on page 210

The second program—call it LONUM—is read in using CLOAD. The original location of HINUM may be written over in whole or part, but it doesn't matter; there's still a copy of the program in upper memory, thanks to the shift. After LONUM is loaded, exit to the monitor once again. Start the second phase of MERGE with the monitor command GO 0035.

MERGE searches LONUM, locates its final 0000 link pointer and calculates the length of the program. If there isn't room in memory for both LONUM and HINUM, the routine terminates with an appropriate abort message. Given sufficient room, HINUM is shifted again, this time so that the first byte of its first header overlays the final 0000 link pointer of LONUM. The programs are now physically contiguous, but changes to HINUM's headers are needed for operational continuity.

Each of the link pointers in HINUM was based on a program starting address of 01D5H. Now, though, HINUM's first header is at a higher location, following all of LONUM. The problem is corrected by adding a constant offset value to each link pointer in HINUM, up to the final 0000 pointer. Since MERGE had earlier calculated the length of LONUM, the offset value is already known.

The final step is to update the address pointers in BASIC's control area. All three pointers are set to the address two bytes after the final zero-value link pointer of HINUM. MERGE prints a final message and returns to the monitor.

The two programs are now fully combined, and will run as a single routine. Alternatively, MERGE can be used over and over to add other routines to the group. Each added routine, remember, must have lower line numbers than any part of the already-merged routines.

MERGE obviously isn't restricted to use with RENUM. Any two or more routines—subject to the line number restrictions—can be combined. A library of useful subroutines can be created, then added as necessary to other programs. The contents of such a library would vary according to need, but might include routines for graphics, data reduction or similar operations that are useful in more than one application.■


SEND FOR FREE CATALOG
OVER 150 ITEMS
Ahoy Dealers!
CALL OR WRITE TODAY FOR YOUR FREE DEALER PACKAGE!
SOFTWARE AUTHORS!
We want to sell your programs!
SEND YOUR TAPE OR DISK (with any documentation) TO OUR SOFTWARE REVIEW BOARD - Dept. R
Adventureland
Pirate Adventure
Mission Impossible
Voodoo Castle
The Count
Strange Odyssey
Mystery Fun House
Pyramid of Doom
Ghost Town
TRS-80 Serious Software
PECA
Savage Island
Poker Tournament
Balrog Sampler
Stone of Sisyphus
Tunnels of Fahal
Star Scout
ATARI 400 & 800
Deflection
Special Bonus Simon Says
Missile Attack
Golf
With Sound Effects
Android
Beewary
Life Two
Kid Venture
Space Colony
Capture
Galactic Saga
Tank Command
Owl Tree
The Amazing Blackjack Machine
Project Omega
Adults Only
Strip Concentration & Dice
Musical Yatc
The Mean Checker Machine
Multi Skill
by Lance Micklus
Star Trek 3.5
Slag
Slag 'Em
High Resolution Graphics Super Sound Effects
Planetoids
Sound
More for your TRS-80
Treasure Quest
Shark Attack
Zoss in Space
Silver Flash
Lunar Lander
Sound for your TRS-80
Word Challenge
Scramble
Interactive Fiction
Real Time "Arcade" Games With Sound Effects
Frog
Ages 5 to Adult
Dr. Chips
ATARI 400 & 800
Angle Worms
Crolon Diversion
ai
Adventure INTERNATIONAL
SOFTWARE PUBLISHING DIVISION OF Scott Adams INC.
ADVENTURE INTERNATIONAL BOX 3435 LONGWOOD, FL 32750 (305) 862-6917

How data is organized, manipulated and stored in memory is must reading for every programmer.

The Wonderful World Of Data Structures

By Phillip J. Windley

Most people view the computer from the perspective of the central processing unit and the functions it performs. There is, however, another important aspect of the computer that is commonly overlooked, especially by those familiar only with high-level programming languages such as BASIC and FORTRAN. That is the manipulation and storage of data in memory—data management.

High-level languages can implement underlying data management structures, but machine- and assembly-language programming determines their basic form.

Data Structures

Data structures form the basis of data management. They are collections of organized data that are in some way related; e.g., telephone books, encyclopedias and multiplication tables. The organization and characteristics of a data structure depend on its function. Convenient access, sequential location, alphabetic storage and easy insertion and deletion of data are qualities we may require of a particular data structure. A filing cabinet, for example, is usually organized alphabetically for convenient access, and is built to allow insertions and deletions. A telephone book, while arranged alphabetically for easy access of data, makes alphabetical insertion of new names and numbers difficult.

Data structures significantly affect the size and speed of a given program. Harold S. Stone, in his book *Introduction to Computer Organization and Data Structures*, says that memory requirements for a program can vary by a factor of two to ten for different organizational representations of the same data. Processing speeds can vary by even larger factors. Choosing the correct data structure saves not only valuable memory space (three cents a byte isn't cheap when you need another 4K), but also a lot of valuable time. Saving ten minutes on a long program is not at all impossible; if you run the program 100 times, that adds up to 16 hours and 40 minutes.

Probably one of the simplest data structures is the one-dimensional array. A one-dimensional array can be compared to a row of books on a shelf. Each book can be identified by its position on the shelf, fifth from the left end, for example. Books can be removed from or placed on the shelf randomly.

Arrays are implemented as consecutive lists in memory and, just as in the book example, each piece of data can be identified by a unique integer corresponding to its position in memory. This unique integer is called the subscript. Input to and output from the array can be completely random.

Implementing arrays in assembly-language programming is fairly straightforward. Set the array's upper and lower bounds and specify a base address at some point in memory prior to using the array. Storing and recalling data is then carried out by two subroutines, the subscript being passed to these subroutines as a parameter. Generalized flowcharts for an algorithm to store data (STORE(N)) and recall data (RECALL(N)) are shown in Fig. 1; N represents the subscript.

In both subroutines the subscript is compared with the upper and lower

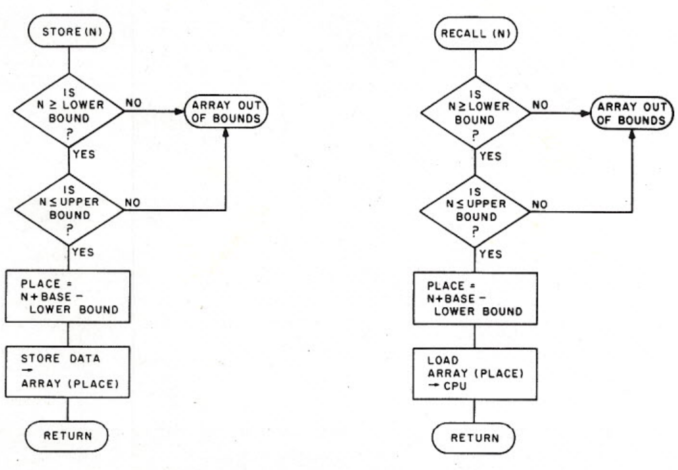


Fig. 1. Flowcharts for STORE(N) and RECALL(N).

Address correspondence to Phillip J. Windley, University of Idaho, College of Mines, Moscow, ID 83843.

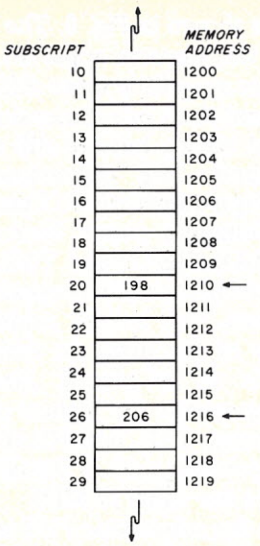


Fig. 2. Configuration of array in memory with examples of STORE and RECALL operations.

bounds of the program to prevent inadvertent damage to data already in memory. If N is greater than the upper bound or less than the lower bound, the array is out of bounds and you'll have to make adjustments. After the bounds have been checked, the address in memory is computed by adding N (the subscript) to the base address and subtracting the lower bound. The data is then stored to or recalled from that memory location.

As an example, say that the lower bound is 10 (all numbers are decimal values), the upper bound is 29 and the base address in memory is 1200. The upper and lower bounds define an array of 20 elements. If the subroutine STORE(N) is called and the parameter N = 20 and DATA = 198 is passed to the subroutine, the computer would first compare the subscript 20 with the lower bound of 10 and the upper bound of 29, find the subscript in bounds and proceed to compute the address of element #20 in the array. It does that by adding the base address 1200 to the subscript 20 and subtracting the lower bound of 10. The result of this operation is 1210. The data, 198, is then stored in memory location 1210 through some form of indirect addressing. As shown in Fig. 2, this is indeed the correct location in memory.

Assume that data has previously been stored in element 26 and you wish to recall that data. When you call the subroutine RECALL(N), the subscript 26 passes to the subroutine. The subroutine, finding 26 to be in bounds, calculates the address in memory of element 26 by adding the base address 1200 to the subscript 26 and subtracting the lower bound 10. The result is 1216, and the data in that memory location is moved to the CPU.

Arrays have a number of uses; one common use of a one-dimensional array is to sort and reorder lists. The bubble sort makes good use of arrays and is an excellent programming example. If you aren't familiar with this sort, it is a good idea to look it over. It can be found in most introductory programming books.

You can also use arrays to store and recall data associated with a particular number. The population figures for the United States between the years 1960 and 1979, for example, can be stored in a one-dimensional array having a lower bound of 1960 and an upper bound of 1979.

Although arrays are handy for sorting ordered lists, they do have some limitations. Inserting data into the array, for instance, is difficult. Let's return to the bookshelf example. If you have a row of books on a shelf and you want to insert another book in the middle, the books already on the shelf must be shifted in some way. Shifting 100 pieces of data up or down one space in memory can be very time-consuming. Later I will describe a type of data structure called a linked list which solves this problem.

Stacks

Stacks are special one-dimensional arrays; rather than random input and random output, storage and retrieval are on a last in, first out basis (LIFO). This type of data structure is analogous to a stack of cafeteria trays; the last tray put onto the stack is the first one taken off. The operations associated with the stack are the POP instruction that pops data off the stack and the PUSH instruction that pushes data onto the stack. In many microprocessors these are part of the instruction set; this makes the stack easy to implement. Even if this is the case, you should know how a stack operates so that you can use it more effectively.

Implementing stacks in assembly language is easy. A generalized flowchart for an array is shown in Fig. 3. In this program there are three values that must be defined before the subroutine is called. Those values are MAX, the highest address in memory used by the stack; MIN, the lowest value in memory used by the stack; and POINTER, the address of the last piece of data pushed onto the stack.

The pointer can either start at the top of the stack and work down the array, or it can start at the bottom and work up. You will notice that I chose to start at the top.

Look at the portion of the main program shown in Fig. 3a and you will see that POINTER is initialized equal to MAX + 1. To push data onto the stack, call the PUSH subroutine (Fig. 3b) and pass the data to be pushed as the only parameter. First the pointer is compared with the value of MIN. If it's less than or equal to MIN, the stack has reached its limits and no new data can be pushed onto the stack. If this is the case, the program will probably let you know that the stack is out of bounds.

If the POINTER is greater than the value of MIN, one is subtracted from the value of POINTER and the data is stored in the memory location addressed by the pointer. The computer then returns to the main program.

When the POP subroutine is called (see Fig. 3c), the pointer is checked to see if it is greater than the value of MAX. If it is, the stack has underflowed (i.e., popped more values

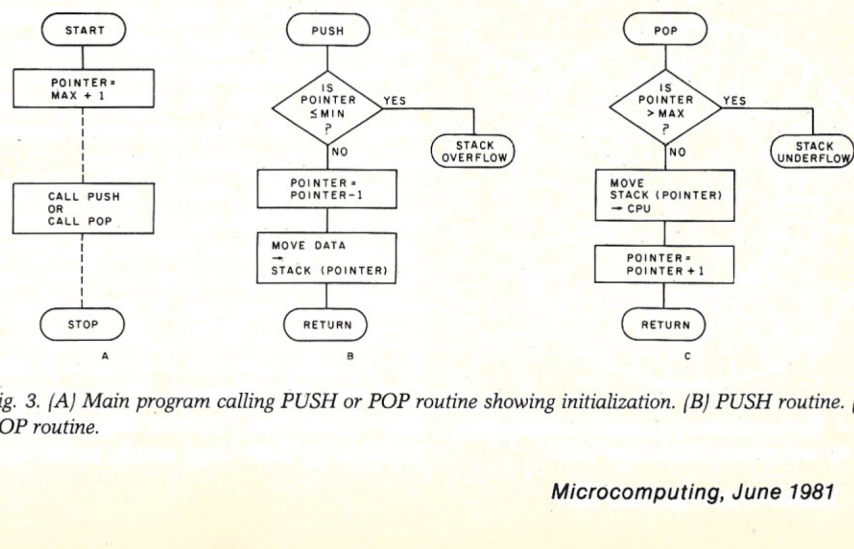


Fig. 3. (A) Main program calling PUSH or POP routine showing initialization. (B) PUSH routine. (C) POP routine.

than were pushed) and the program is in error. If not, the computer moves the data byte addressed by the pointer into the CPU. Following that, one is added to the pointer so that it points to the next byte of data in the stack, and the computer resumes execution of the main program.

As an example of program execution, assume that MAX has a value of 4019 and MIN has a value of 4000. During initialization, POINTER would be set to 4020, which is the value of MAX+1 (shown by the dotted arrow in Fig. 4a). When the subroutine PUSH is called, the value of POINTER is compared to the value of MIN. Since 4020 is greater than 4000, the program proceeds. One is

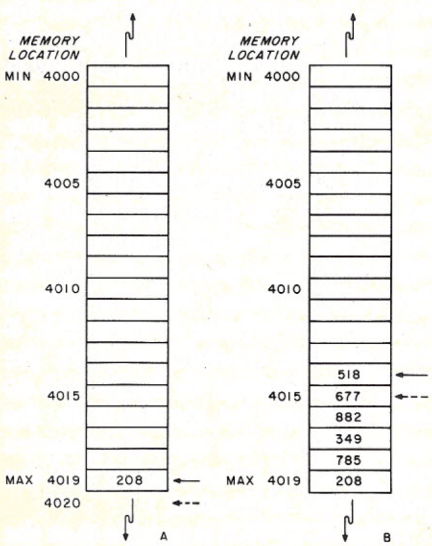


Fig. 4. Operation of a stack in memory. (A) Dotted arrow shows the position at initialization; solid arrow represents the pointer after a PUSH operation has been called. (B) Solid arrow represents the pointer at the end of the PUSH operation storing 518; dotted arrow is the position of the pointer after 518 has been POPPED.

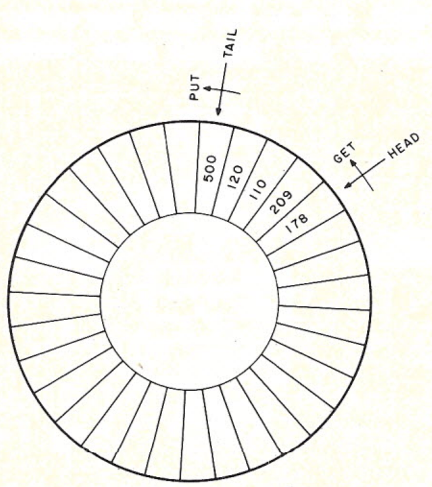


Fig. 5. Representation of a queue in memory showing the two pointers HEAD and TAIL.

subtracted from the pointer, giving a new value of 4019. The data to be PUSHED onto the stack, 200, is moved to memory location 4019, and the computer resumes execution of the main program.

Now, assume that five other numbers, 785, 349, 882, 677 and 518, have been pushed onto the stack. The value of POINTER is now 4014 (see Fig. 4b).

Let's look at the execution of the POP subroutine. When you call the POP subroutine, the value of POINTER (4014) is compared with the value of MAX (4019). Since 4014 is less than 4019, the data stored in memory location 4014 (518) is moved to the CPU and the value of POINTER is incremented so that its value is 4015. The pointer now points at the data that was stored immediately before 518, and the computer resumes execution of the main program.

Stacks are used extensively in assembly-language programming. They are very convenient for temporary storage of registers and for storing return addresses when subroutines are called. A stack is especially useful in servicing interrupts. When a peripheral device interrupts the CPU, the current operation is temporarily suspended and the CPU starts execution of a routine designed to service that particular device. Since the CPU was interrupted in the middle of an operation, the computer must store all the registers. Then, when the CPU has finished servicing the peripheral device, the computer can resume execution of the interrupted operation at the place it was suspended.

Queues

Stacks are good for quick, temporary storage of new data, but they have one quality that is not always desirable; the last byte of data stored is the first to be retrieved. In some programs you might want to temporarily store data and then recall it in the same order in which it was stored. To do this you could use a data structure called a queue.

Queues, like stacks, are special one-dimensional arrays, but instead of a last in, first out (LIFO) structure, the queue uses a first in, first out (FIFO) arrangement. A queue could be likened to a dixie cup dispenser—the first cup into the dispenser is the first one out. There is, however, one big difference; queues are usually written so that they seem to be circular, as shown in Fig. 5. The reason for this will become apparent.

Queues are a little more difficult to implement than one-dimensional arrays and stacks, but are still easy to understand. The operations performed by a queue are PUT and GET. To implement a queue, four values, MIN, MAX, HEAD and TAIL, must be defined in the main program. MIN represents the lowest address in memory of the array in which the queue is to be placed. Similarly, MAX represents the highest address of the array. TAIL and HEAD serve as the pointers and at initialization are set to the value of MIN.

When the subroutine PUT is called (refer to Fig. 6a), the value of TAIL is incremented and checked to see if it is in bounds. First, the value of TAIL is compared with the value of MAX. If it *is* greater, the value of TAIL is set to equal the value of MIN (this is what makes the queue circular). If TAIL is *not* greater than MAX (or, if it is, after it has been set to equal the value of MIN), TAIL is compared to the value of HEAD. If they are equal, the queue is full and the appropriate action is taken. If not, the byte of data is stored in the queue using the value of TAIL for an address. The computer then resumes execution of the main program.

The subroutine GET (Fig. 6b) operates in much the same manner as the PUT subroutine. The values of HEAD and TAIL are compared. If they are equal, the queue is empty and the program is in error; if not, one is added to the value of HEAD. Next, the value of HEAD is compared with the value of MAX. If HEAD is greater than MAX, HEAD is reset to the value of MAX (or if it is, after it has been reset to the value of MIN, the data in the memory location addressed by the value of HEAD is loaded into the CPU).

To better illustrate just how a queue works, draw out a set of memory locations as shown in Fig. 7 and label them 100–124. 100 is the value of MIN and 124 is the value of MAX. Take a pencil in your right hand and a pencil in your left hand. Call one of the pencils HEAD and the other TAIL; these will serve as pointers. Now, initialize your pointers to the value MIN by setting one down on each side of memory location 100 with the pencils pointing at it.

Try implementing the PUT routine shown in Fig. 6a. First, the value of TAIL is incremented by one, so move

Orange Micro

"THE COMPUTER PRINTER SPECIALISTS"

UP TO 25% DISCOUNTS! — SAME DAY SHIPMENT!

CENTRONICS 737 (RADIO SHACK LINE PRINTER IV)

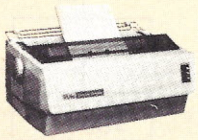
Word Processing Print Quality

• 18 x 9 dot matrix; suitable for word processing • Underlining • proportional spacing • right margin justification • serif typeface • 50/80 CPS • 9½" Pin Feed/Friction feed • Reverse Platen • 80/132 columns

CENTRONICS 737-1 (Parallel)	(List $995)	**$765**
CENTRONICS 737-3 (Serial)	(List $1045)	**$815**

VISTA — C. ITOH

Daisy Wheel Letter Quality



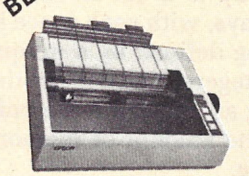
• 25 CPS (Optional 45 CPS) • Typewriter quality • Centronics parallel • RS 232 Serial (Optional) • Proportional spacing • Bidirectional • Programmable VFU • Self test • Diablo compatible • Friction feed (Optional tractors) • 136 printable columns. • Manufactured by C. ITOH.

VISTA V300 (C. ITOH)	(List $1895)	**$ Call**

EPSON MX80/MX70

OUR BEST BUY!

Low-Priced Professional Print Quality



• 9 x 9 dot matrix • Lower case descenders • 80 CPS • Bidirectional, Logic seeking • 40, 66, 80, 132 columns per line • 64 special graphic characters: TRS-80 Compatible • Forms handling • Multi-pass printing • Adjustable tractors

EPSON MX80	(List $645)	**$Call**
EPSON MX 70 Dot graphics, 5 x 7 matrix	(List $450)	**$Call**

ANACOM

Low Cost, High Speed, Wide Carriage

• 9 x 9 dot matrix • Lower case descenders • Wide carriage • Adjustable tractors to 16" • 150 CPS, Bidirectional, Logic Seeking

ANACOM 150	(List $1350)	**$ Call**

ANADEX

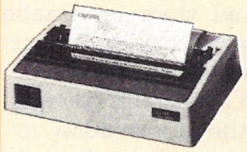
Dot Graphics, Wide Carriage

• 11 x 9 dot matrix; lower case descenders • Dot resolution graphics • Bi-directional, logic seeking • Up to 200 CPS • RS 232 Serial & Parallel • Forms control • X-ON/X-OFF • Up to 6 part copy.

ANADEX 9501	(List $1650)	**$1350**

OKIDATA MICROLINE SERIES

TRS-80 Graphics Compatibility, Friction Feed



• 9 x 7 dot matrix • 80 CPS • 80, 132 columns — 64 shapes for charts, graphs & diagrams • Double wide characters • 6/8 lines per inch • Up to 3 part copy • Friction & pin feed • 200 M character head warranty

OKIDATA MICROLINE 80	(List $800)	**$520**
OKIDATA M82 Bidirectional, Forms handling	(List $960)	**$750**
OKIDATA M83 Wide carriage, 9 x 9 dot matrix	(List $1260)	**$1050**

NEC SPINWRITER

High Speed Letter Quality

• 55 CPS • Typewriter quality • Bidirectional • Plotting • proportional spacing.

5510-5 RO, Serial, w/tractors	(List $2995)	**$2625**
5530-5 RO, Parallel, w/tractors	(List $2970)	**$2599**

TELEVIDEO CRT'S

AT DISCOUNT PRICES!

QUANTITY PRICING AVAILABLE

TVI 912C
TVI 920C
TVI 950

Please Call Toll Free
Prices are too low to advertise

IDS PAPER TIGERS

Dot Resolution Graphics, quality print, speed

• 7 wire printhead (445); 9 wire printhead (460) with lower case descenders • Over 150 CPS • bi-directional, logic seeking (460) • 8 character sizes; 80-132 columns • Adjustable tractors • High-resolution dot graphics • Proportional spacing & text justification (460).

IDS 445G 7 wire printhead, graphics	(List $895)	**$ 750**
IDS 460G 9 wire printhead, graphics	(List $1394)	**$1150**
IDS 560G 9 wire, wide carriage, graphics	(List $1794)	**$1590**

PRINTERS

MALIBU 165 wide carriage, graphics, letter quality	(List $2495)	**$ 1975**
QUME 5/45 typewriter quality	(List $2905)	**$ 2559**

INTERFACE EQUIPMENT

CCS APPLE PARALLEL Interface & cable		**$ 150**
APPLE II - EPSON MX80 parallel interface board & cable		**$ 100**
SSM AIO BOARD Apple Serial/parallel interface	(List $225)	**$ 175**
MICROTRONICS Atari parallel interface		**$ 69**
ATARI 850 Interface module, serial/parallel		**$ 199**
TRS-80 CABLES to keyboard or Exp. interface		**$ Call**
NOVATION D-CAT direct connect modem		**$ Call**

CALL FOR FREE CATALOG

(800) 854-8275

CA, AK, HI (714) 630-3322

At Orange Micro, we try to fit the right printer to your application. Call our printer specialists for free consultation.

TELEPHONE ORDERS: Mon.-Fri. 8:30 - 5:00
The Orange Micro Printer Store (Retail):
Mon.-Fri. 10:00 - 6:00, Sat. til 4:00

Phone orders WELCOME; same day shipment. Free use of VISA & MASTERCARD. Personal checks require 2 weeks to clear. Manufacturer's warranty included on all equipment. Prices subject to revision.

Orange Micro, Inc.
3150 E. La Palma, Suite I
Anaheim, CA 92806

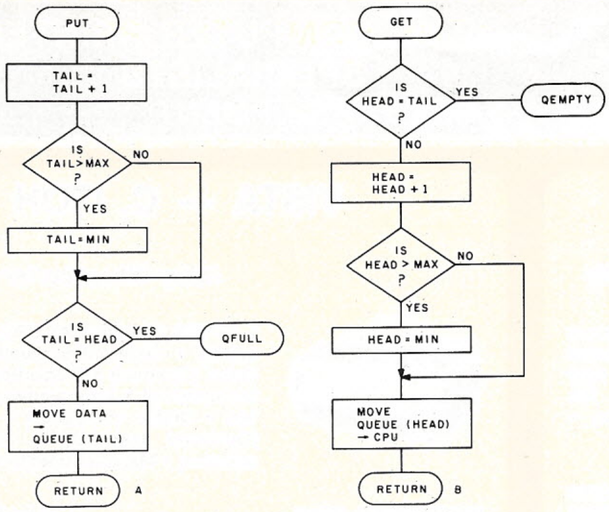


Fig. 6. (A) The PUT operation of a queue. (B) The GET operation of a queue.

the pencil representing TAIL to point at memory location 101. You, acting as the CPU, can see that 101 is not greater than the value of MAX (124), nor is it equal to the value of HEAD. Therefore, data can be stored in memory location 101 without any further processing.

Assume that this happens 23 more times without a GET routine having ever been called; your pencil should move down 23 spaces until it points at memory location 124, the value of MAX. Remember, as your pencil moved down the list, data was stored in each of those 23 locations.

If PUT is called again, without GET having been called, the following will happen. TAIL will be incremented so that it points to memory location 125.

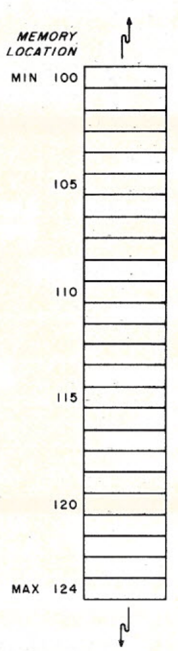


Fig. 7. Representation of memory for simulation of PUT and GET operations.

Since 125 is greater than the value of MAX, 124, TAIL will be reset to the value of MIN. (Move your pencil back to 100.) The pencil representing HEAD should still be there, and the computer (you) can see that HEAD and TAIL are equal, so it branches to the routine QFULL; no more PUT operations can be performed until a GET operation takes place.

Try running through some GET operations on your own. If you play around with this, you will find that your pencils can never pass each other—they only chase each other around the circle. When TAIL catches HEAD the queue is full, and when HEAD catches TAIL the queue is empty.

Queues are very useful in I/O operations. Often, when the CPU is interrupted it is not in a position to process the incoming data immediately. The data can be stored in a queue for subsequent processing. The reverse is also true. As the computer processes data to be output, it is often desirable to store the data in a queue and output it all at once, rather than output the data singly. This saves CPU time and also saves wear and tear on peripheral equipment such as disks, which are only engaged when involved in an actual I/O operation.

Advanced Data Structures

One-dimensional arrays, stacks and queues are simple but useful data structures. There are other, more complex data structures requiring programs of greater difficulty than those discussed so far. They are usually generalized cases of the data structures I have already outlined. A good understanding of arrays, stacks and queues aids the programmer dealing with these more complex data structures, so I will touch lightly on them as an introduction but leave the actual programming to another time.

The first of these more advanced data structures may already be familiar—the multidimensional array. Remember that a one-dimensional array is implemented as a consecutive list in memory. A two-dimensional array is a set of one-dimensional arrays, the actual number of arrays being given by a second subscript. A two-dimensional array is somewhat like a chessboard, a set of eight rows of eight. Each space on the board can be described by a unique set of two integers. For instance, the set of integers (2,3) could describe the third space on the second row. A three-dimensional array is a set of two-dimensional arrays, with each space in the array being defined by a unique set of three integers. You can continue this as long as you want; the only limitation is the amount of memory space available.

Multidimensional arrays are useful in modeling the world around us because the world is multidimensional. Say, for example, that you wanted to write a program to take care of ticket reservations for a local theater. Since the seats have row and column numbers, you could set up a two-dimensional array and store information about each seat (occupied, unoccupied or unavailable) in the array. If the theater had a balcony, you might want to put the information in a three-dimensional array, since you have a set of two two-dimensional arrays.

There are many other uses for them. If you are an electronics buff, circuit analysis can entail working with matrices. Multidimensional arrays make matrices a snap!

Deques

Another type of data structure is the double-ended-queue or deque (pronounced "deck"). A deque is a hybrid version of a stack and a queue. In the queue we discussed earlier, the subroutine GET took data out of the queue at the pointer HEAD, and the subroutine PUT stored data at the pointer TAIL. In the deque, data can be put or got at either the HEAD or the TAIL. Four routines, PUTTAIL, GETTAIL, PUTHEAD and GETHEAD, are used. If you use only the routines GETTAIL and PUTTAIL, you can use the deque as a

stack. If you use only the routines PUTTAIL and GETHEAD, you can use the deque as a regular queue.

Deques are useful in specialized manipulations of data. They are also useful in saving memory space, because the same array space can be used as a queue or a stack, depending on the current needs of the program.

Linked List

The last type of data structure I will talk about is the linked list. Earlier, I mentioned that it was difficult to insert data into the middle of an array without shuffling a lot of things around in memory. A linked list solves this problem.

Linked lists are somewhat like the treasure hunts we used to play at as children. Each group of children is given a note with a clue to the location of the next clue. That clue, in turn, gives the location of the next clue. This continues until the treasure is found.

In linked lists there is no treasure, but the idea is the same. Associated with each byte of data is the address of the next byte in the list. Because of this, linked lists don't have to occupy consecutive locations in memory. To insert a byte of data into the list, the address portion of the data byte preceding the new data is modified to point to the new data byte, and the address of the data byte following the new data is written into the address portion of the new data. To delete data, the procedure is reversed.

I think you can see, however, that linked lists do require quite a bit more memory space to store the addresses. Linked lists are used in handling text, sorting lists and other tasks where data must be inserted and deleted often.

Conclusion

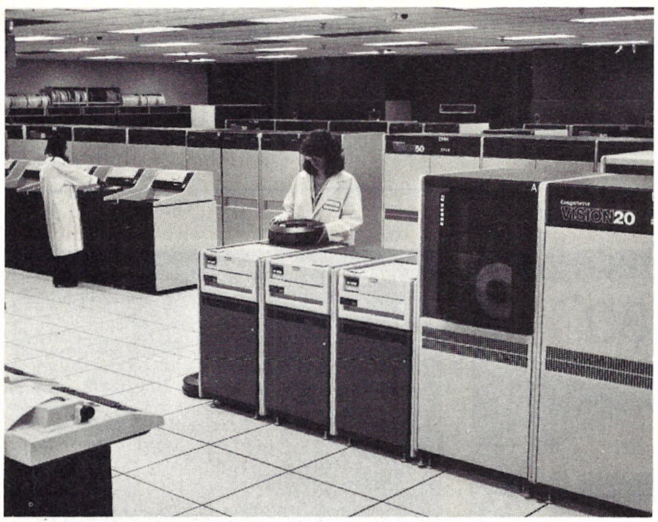
These are just a few of the data structures being used in computer programming today; there are many more. This brief introduction should serve to get you started. Many books have been written on the subject and can give you a more detailed explanation of the structures I have talked about, as well as explanations of many more I haven't.

Familiarity with data structures is important if you want to become more efficient in your programming techniques. Put them to work, and they will save you a lot of memory for program luxuries and a lot of time so that you can enjoy them. ■

✓See List of Advertisers on page 210

We've lowered the

No, the low CompuServe $5.00 per hour charge hasn't changed, but the world of information we offer is growing steadily.



Of CompuServe's 22 large computer systems, 10 are housed here in our Dublin, Ohio Computer Center.

More Computer Power

The number of CompuServe customers keeps growing—now more than 8,000 strong. The more customers we have, the more computer power and access lines we dedicate to our Information Service. Our 22 mainframe computer systems provide sufficient power and capacity to serve thousands more.

More News

In addition to the Associated Press, we've added more news. Read The New York Times and all the news that's fit to print, including Broadway show and movie reviews, stories from Pulitzer Prize-winning writers and more. We are continually adding new features to the information service, so check our "What's New" section regularly to see what other information is available.

TRS-80™ Color Computer and Videotex™

The CompuServe Information Service is sold in Radio Shack® Stores and, in addition to all the major brands of computers and terminals, can be used with the new TRS-80 Color Computer and Videotex terminal. Our special software formats the information in easy-to-read pages. Go forward, backward, anywhere in the CompuServe Information Service databases—even directly to a page of information.

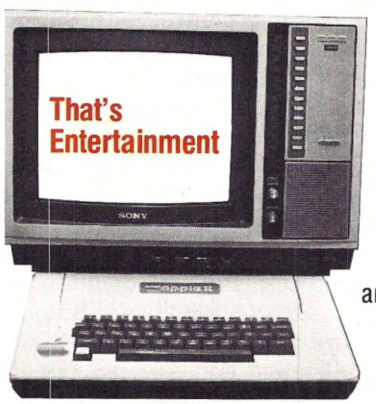
Big System Reliability

When we say we're reliable, we mean it. In fact CompuServe computers were up and running for a 99.6 percent reliability factor last year. We've also built in many "invisible" features like data error detection and retransmission—all those technical enhancements which add up to the CompuServe Information Service being ready to use when you're ready to use it.

cost of the world.

Access to all the CompuServe services, including electronic mail; bulletin board; CB simulation; newsletters from Radio Shack®, Atari®, and others; up to 128K storage free plus all the services listed here are yours. Access time is only 8⅓ cents per minute between 6 PM and 5 AM weekdays and all day weekends, billed to your charge card. It's a local phone hook-up in more than 260 U.S. cities.

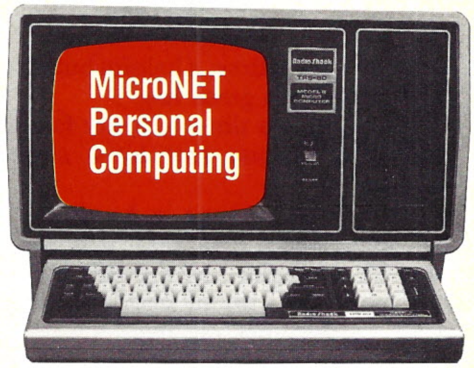
There's an energy management system, home repair tips, discount information, personal health information and more. Better Homes & Gardens provides household information such as recipes, nutritional analyses and meal plans.



All kinds of games (including some for the kids), book and movie reviews, astrology and the latest sporting event news — even point spreads and injury lists. There's the CompuServe line printer art gallery and even a national user opinion poll for instant feedback on current events.

A complete program. The AP financial wire, newspaper financial pages, Raylux Financial Advisory Service, the Commodity News Service and personal financial programs. At a slight additional charge, you can retrieve fundamental financial data on more than 1,700 companies from Value Line, and MicroQuote offers current and historical data and special market reports on more than 40,000 stocks, bonds and options, updated daily.

While the CompuServe Information Service is designed for everyone, MicroNET Personal Computing is for the knowledgeable computer hobbyist or business person. You can buy software and have it downloaded directly to your person computer. Use programming languages such as *APL-SF, EXTENDED BASIC, MACRO-10, SNOBOL, EXTENDED FORTRAN* and *PASCAL,* use word processing and text editing programs.



Not 1985, NOW!

See for yourself what a state-of-the-art electronic information service can do. Get a demonstration at a Radio Shack® computer center or store — or write to us for further information.

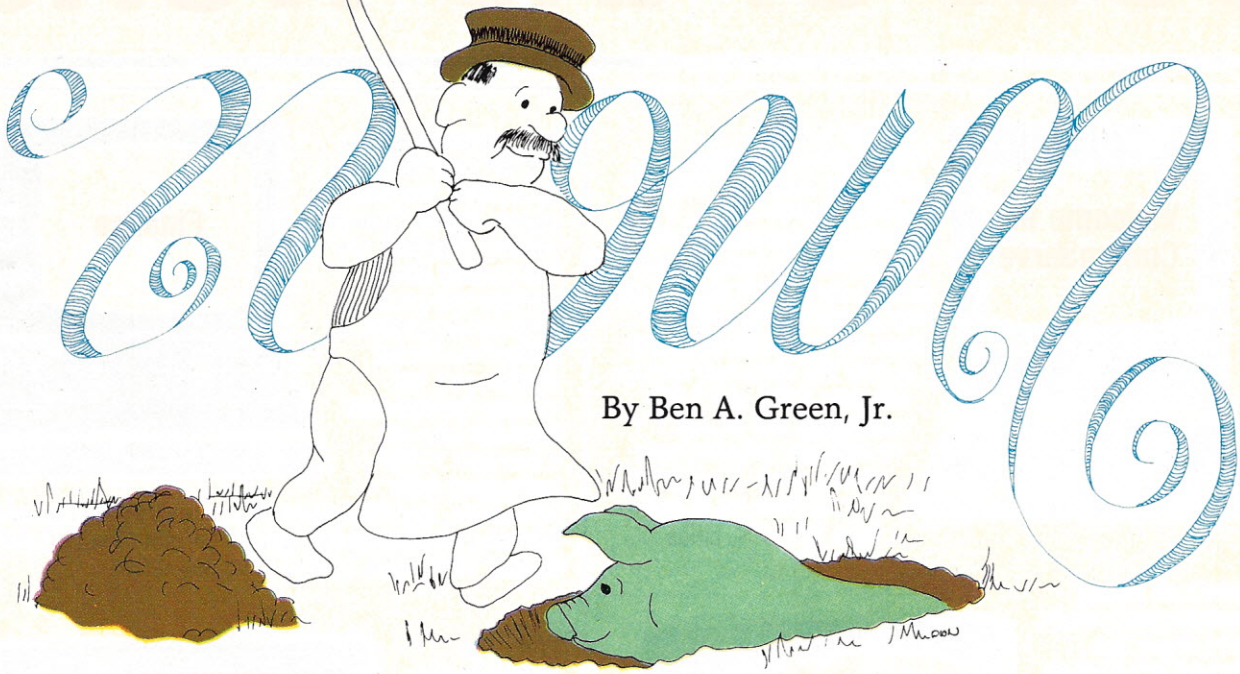
CompuServe

147

Information Service Division
5000 Arlington Centre Blvd.
Columbus, Ohio 43220
(614) 457-8600

Write your own ending with this exercise in recursive processes for your PET.

Once upon a Time . . . .



By Ben A. Green, Jr.

"Once upon a time, a helpful butcher dug up a green hog."

"Once upon a time, a thrifty fakir ignored a tortoise that hit a sour crooner."

"Once upon a time, a handsome pelican bribed a royal fakir that dug up a kind burglar that a pretty king that embraced a nanny beside a cowardly tortoise was bugging."

I didn't make these up; my PET did, using the program Fable described in this article. When the program was finally finished, my family and I spent all afternoon laughing ourselves silly over the output, so I thought you *Microcomputing* readers might enjoy it, too.

The idea for Fable came from Douglas Hofstadter's marvelous book *Goedel, Escher, Bach* (Basic Books, 1979), a deep, funny, stimulating study of artificial intelligence. One of its central concepts is the recursive process, whose execution involves itself as a part. In computer applications an example is a subroutine that calls itself.

A subroutine that calls itself might never end; it might be caught in an endless loop. But if there is at least one possible control path through it that does not call the subroutine itself, it will terminate eventually, if the probability of taking the non-calling path is not zero.

To illustrate his idea, Hofstadter refers to a process called "fancy noun," which calls itself recursively, and "ornate noun," a simpler, nonrecursive process. Both are diagrammed in Fig. 1. Ornate noun has a straight-line control path that produces an article, an adjective and a noun, but it can also skip the article and either skip or repeat the adjective.

Fancy noun starts by calling ornate noun, but then takes one of three possible control paths: one that goes

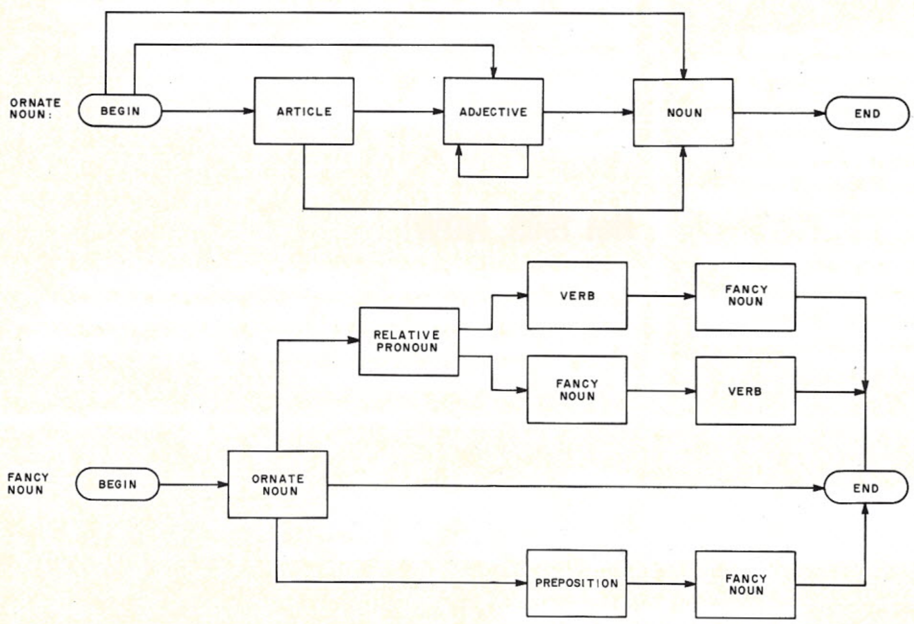


Fig. 1. Control paths for ornate noun and fancy noun.

Address correspondence to Ben A. Green, Jr., 2370 Algonquin Road, Schenectady, NY 12309.

```
100 REM===========FABLE==============
110 :
120 REM-----READ DATA-----------------
130 :
140 N=30
150 DIM NOUN$(N),ADJ$(N),VERB$(N),PREP$
(5)
160 FORI=1TON:READ A$:NOUN$(I)=A$+" ":N
EXTI
170 FORI=1TON:READ A$:ADJ$(I)=A$+" ":NE
XTI
180 FORI=1TON:READ A$:VERB$(I)=A$+" ":N
EXTI
190 FORI=1TO5:READ A$:PREP$(I)=A$+" ":N
EXTI
200 DEF FNA(X)=INT(X*RND(1)+1)
210 :
220 REM=====MAIN PROGRAM=============
230 :
240 PRINT"■ONCE UPON A TIME, ";
250 GOSUB330:REM:ORNATE NOUN
260 PRINT VERB$(FNA(N));
270 GOSUB420:REM:FANCY NOUN
280 PRINT"■.

290 PRINT"■■PRESS SPACE TO CONTINUE."
300 GET A$:IF A$="" THEN300
310 GOTO240:REM:REPEATS INDEFINITELY
320 :
330 REM---SUBROUTINE 'ORNATE NOUN'---
340 :
350 PRINT"A ";
360 IF RND(1)<.2 THEN390
370 :PRINT ADJ$(FNA(N));
380 :IF RND(1)<.05 THEN PRINT"■, ";
    :GOTO370
390 PRINT NOUN$(FNA(N));
400 RETURN
410 :
420 REM---SUBROUTINE 'FANCY NOUN'----
430 :
440 GOSUB330:REM:ORNATE NOUN
450 ON FNA(4) GOTO480,510
460 RETURN
470 :
480 PRINT PREP$(FNA(5));
490 GOTO420:REM:RECURSION & EXIT
500 :
510 PRINT "THAT ";
520 IF RND(1)<.3 THEN560
530 :PRINT VERB$(FNA(N));
540 :GOTO420:REM:RECURSION & EXIT
550 :
560 GOSUB420:REM:RECURSION & RETURN
570 PRINT VERB$(FNA(N));
580 GOTO460
590 :
600 REM========DATA=================
610 :
620 REM---NOUNS:---------------------
630 :
640 DATA BOY,CAT,COW,GIRL,HOG,JACKRABBI
T,LADY,MOUSE,NANNY,POSSUM
650 DATA QUEEN,ROBOT,SINGER,TIGER,KING,
BURGLAR,CROONER,DOG,FAKIR,GROCER
660 DATA POLICEMAN,SOLDIER,BAKER,BUTCHE
R,CONGRESSMAN,COOK,PELICAN,ROBIN
670 DATA "CANDIDATE FOR PRESIDENT",TORT
OISE
680 REM---ADJECTIVES:---------------
690 :
700 DATA RICH,GREEN,DINGY,FOUL,GEM-BEDE
CKED,HAPPY,KIND,LAZY,STINGY
710 DATA STRANGE,PRETTY,POOR,MOROSE,DEA
F,TACKY,VAIN,WICKED,YOUNG,CLUMSY
720 DATA COWARDLY,BRAVE,HANDSOME,ROYAL,
SENILE,LOYAL,FRIENDLY,COURTEOUS
730 DATA HELPFUL,LIBERAL,CONSERVATIVE
740 :
750 REM-----VERBS:------------------
760 :
770 DATA BURNED,CRUMPLED,HIT,FOUND,"GOB
BLED UP",HUNG,KILLED,LOVED,MET
780 DATA TICKLED,"WAS BUGGING",STABBED,
"DUG UP",ATE,SHOT,GREETED,HUGGED
790 DATA SLAPPED,SAW,INSULTED,"SAT ON",
SCRATCHED,KISSED,HATED
800 DATA"SPAT ON",EMBRACED,"JUMPED ON",
CHEATED,BRIBED,GOOSED
810 :
820 REM----PREPOSITIONS:------------
830 :
840 DATA FOR,WITH,"BECAUSE OF","NEXT TO
",BESIDE
```

Fable program.

straight through to the end, a low one that adds on a prepositional phrase or a high one that adds on a dependent clause like "that the boy hit" or "that hit the boy." But in place of simply "the boy," fancy noun calls for a fancy noun in that place. This also happens in the low path, where instead of merely "with a pelican," for example, the process calls for a fancy noun to put after "with." This feature makes fancy noun a recursive process. Each call to fancy noun will call upon fancy noun until the straight path is taken, which terminates the process.

The program Fable uses these two processes as subroutines (I modified ornate noun to make it always choose the article "a," which seemed most appropriate for a fable) to build a complete sentence beginning "Once upon a time." It uses the PET's random number function RND(1), which produces a decimal fraction between 0 and 1 with a uniform probability distribution, to make the choice of control path and of vocabulary whenever necessary. After loading the arrays with 30 nouns, adjectives and verbs and five prepositions, and defining a function to produce random integers between 1 and X, it is ready to write fables. It starts with an ornate noun and a verb. Then it gets a fancy noun to serve as the object of the verb. That's it.

Lines 360 and 380 set the probabilities for skipping or repeating adjectives. The probabilities for the three branches of fancy noun are set in line 450: if FNA(4) returns 1, we get a prepositional phrase; with 2 we get a dependent clause; and with either 3 or 4 we simply fall through to line 460, which represents termination of the subroutine. (If you change line 450 to include FNA(2), then the sub-

routine will never terminate and you will get a sentence that never ends.) In the "that" branch, the choice between "that hit the boy" and "that the boy hit" is made in line 520.

You can, of course, substitute your own vocabulary; some of the examples I have quoted were produced from earlier versions. It might be interesting also to fix the program so that it will not repeat a noun.

The sentences can get complicated. For a while I was stumped by "Once upon a time, a trustworthy queen found a loyal butcher that gobbled up a sour jackrabbit that a dingy tiger that a young policeman that fixed a vain mouse that scratched a morose tiger over a trustworthy grocer was seeing crumpled." Eventually, I figured out that it was the policeman who was seeing the dingy tiger, who, in turn, was the one who crumpled the rabbit.

My favorite fable, though—one that has a lot to teach us—is the following:

"Once upon a time, a brave king hit a clumsy soldier with a deaf, stingy, dingy jackrabbit."

I'm still wondering why. ■

You must be twice as determined to operate the break key with this simple circuit.

Double-Good OSI Protection

By Geoff A. Cohen

This circuit eliminates accidental operation of the break key on my OSI C1. The key has to be pressed *twice* within one second.

I installed an LED in the break key, in a similar fashion to the shift lock on DEC machines (see Fig. 1). Note that this device is placed in series with the line between the break key and the computer reset pin (6502 pin number 40).

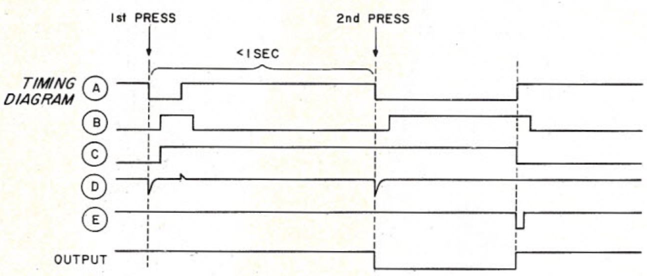
Circuit Operation

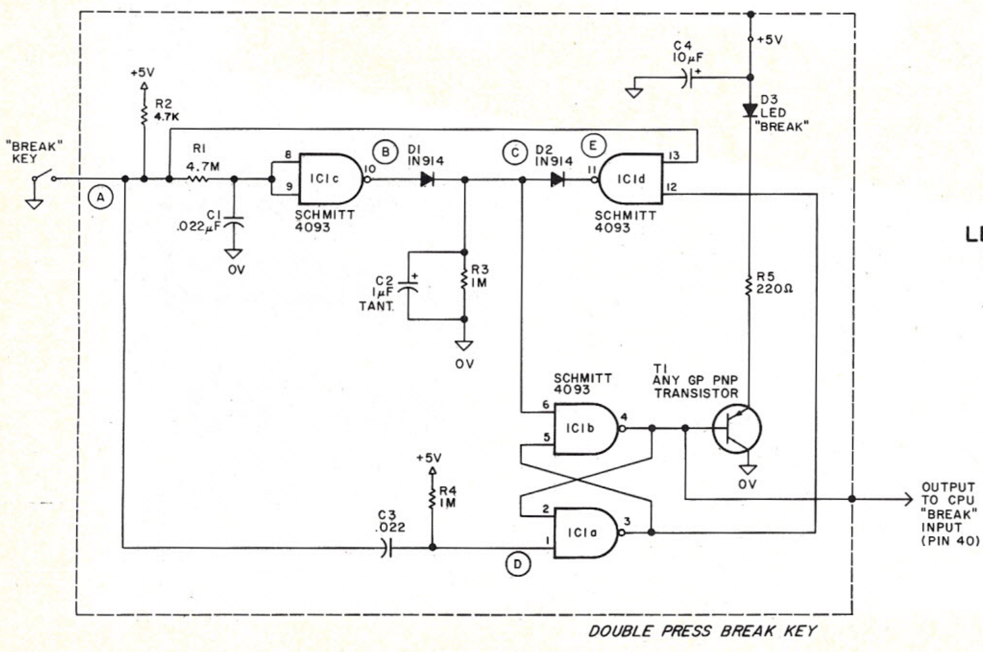
When the break key is first pressed, the output of IC1c goes high after 100 ms delay (R1 + C1), charging the one second timeout capacitor (C2) via diode D1. This allows the RS flip-flop (IC1a + b) to be triggered by an input pulse to pin 1.

If the break key is pressed again within one second of the first press, a short (20 ms) negative pulse (via R4 + C3) sets the RS flip-flop (IC1a + b) output to low. The output will stay low as long as the break key is pressed. When the break key is released, IC1d will discharge the timeout capacitor (C2) through diode D2.

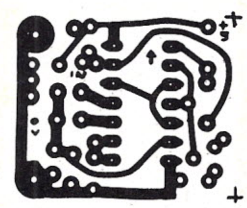
The device is now ready for the sequence to be repeated when the break key is again pressed. ■

Address correspondence to Geoff A. Cohen, 72 Spofforth St., Holt, A.C.T. 2615, Australia.

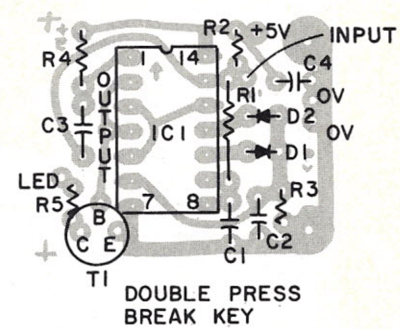




Circuit and timing diagram.



PC board layout.



Component layout.

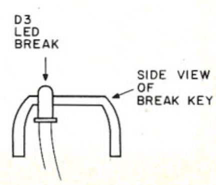


Fig. 1.

NEVER UNDER$OLD.

Thats right. If you can find a lower price in this magazine WE WILL BEAT IT! Period.*

S-100 HARDWARE

SD SYSTEMS
- Versa Floppy I kit $259
- Versa Floppy II kit $349
- ExpandoRam II kit $249

ITHICA INTERSYSTEMS .... $CALL
ALTOS .... $CALL
NORTHSTAR .... $CALL

SUPPLIES

Verbatim Datalife Diskettes
MD525-01,10,16 .... $26.50
FD34-9000 .... $30.00
FD34-8000 .... $44.00

Avery Tabulables
5,000 3½ X 15/16 ...... $18.75
3,000 3½ X 15/16 ...... $14.25
1,000 3½ X 15/16 .... $8.15

UARCO Paper (prices are FOB S.P.)
9½ X 11, 15 or 18 Lb. $25.95
14 7/8 X 11, 15 or 18 Lb. . $35.00

SOFTWARE

Microsoft Basic-80 .... $299
Microsoft Basic Compiler $319
Microsoft Fortran-80 ...... $399
Magic Wand $275
Apple Visicalc $113
Pickles and Trout CP/M® $175

APPLE HARDWARE

Microsoft Z-80 Softcard . $259
Microsoft Ramcard $170
SSM AIO A&T $165
SYMTEC Apple Light Pen . $219
MOUNTAIN HARDWARE:
- Super Talker $270
- ROMWRITER $157
- INTROL/X-10 .... $180
- ROMPLUS + ... $162
- MUSICSYSTEM $499
- Apple Clock $252

Lobo Drive .... $CALL
Videx 80 x 24 .... $299
Andromeda $170
M & R Sup-R-Term $299
Enhancer $100
NEC Green Monitor $240
Sanyo Green Monitor $240
D.C. Hayes Micromodem . $299
CALIFORNIA COMPUTER SYSTEMS
See our full page ad in this magazine.

PRINTERS

EPSON MX-80 .... CALL
MX-80 Graphics ROM ... CALL
NEW EPSON MX-80 TF .. CALL
Okidata Microline 80 .... $415
Okidata Microline 82 .... $625
Okidata Microline 83 .... $899
IDS Paper Tiger 460G .. $1050
Anadex DP-9000,1 .... $1195
Anadex DP-9500,1 .... $1295
Centronics 737 .... $689
NEC Spinwriter .... $CALL
NEC w/ Sellum Option ... $CALL
C. Itoh Starwriter 25 cps $1440
C. Itoh Starwriter 45 cps $1770
Diablo 630 .... $2195
Malibu 165 .... $1995
Malibu 200 .... $CALL
MPI 88G $495

TRS-80 HARDWARE

TRS-80's .... $CALL
Micropolis 77 track $399
Percom Doubler $199
Percom Separator $27
Orchestra 80 .... $79

ALPHA BYTE STORES ✓311

5115 Douglas Fir Road • Suite B • Calabasas, CA 91302

*Include $2.00 shipping and handling on all orders. Visa and Master Card accepted. Never undersold offer applies only to items currently deliverable from other retail vendors at advertised price. All never undersold offers good as supply lasts. Please add $2.00 for all COD orders. Please call for items not listed. We gladly answer any questions on all of our hardware, software, and supply needs. Quanitity discounts available. School purchase orders accepted. Please remember to figure competitors shipping and handling charges when arriving at never undersold price.

(213) 883-8594

Here's a program that helps you find your way from hex to inverted decimal and back again.

To and Fro with Apple's Inverted Decimal Code

By Don Lancaster

The Apple II, like many other microcomputers, has two ways of calling out locations in its address space.

If you are using assembly or machine language, the 65536 locations in the address space are called out with a hexadecimal code, ranging from 0000 to FFFF.

If you are using Integer BASIC or another higher-level language, these same address locations are called out as a decimal number, ranging from −32768 to +32767. Decimal numbers are more useful in BASIC because they eliminate the strings you might have to use for the hex A, B, C, D, E and F values that stand for decimal 10 through 15.

The obvious thing to do is to let each decimal location be the same numeric equivalent of each hex location. But this would take numbers that go from 0 to 65535, and Apple's Integer BASIC only allows numbers from −32767 to +32767.

To beat this problem, the Apple people invented an inverted decimal code. It uses the bottom half of the memory as direct numeric equivalents and the top half of the memory as upside down decimal negative values. This is roughly akin to Apple II's complement notation, where some of the numbers seem to be heading the wrong way at first glance. Fig. 1 shows two ways the Apple address space is mapped.

For locations hex 0 to 7FFF, the decimal code is the same as its hex value. Decimal 0 to +32767 goes with hex 0 to 7FFF. For locations hex 8001 through FFFF, the decimal code counts backwards, and with a minus sign. This way, hex FFFF is decimal −1; hex FFFE is decimal −2. The down counting continues to −32767 at hex 8002, and finally −32767 at hex 8001.

Location hex 8000 is complex. I'll return to it later. It is coded as decimal −32768, but you can't get there from here without using some tricks.

Why Convert?

Why would you ever have to worry about these two codes? Any time you write a BASIC program, if you want to turn loose the full power of the monitor, you have to use the inverted decimal code to call whatever it is you are after. If you want to change or look at a memory location in your machine, you also have the PEEK and POKE commands, also called as decimal numbers.

Some locations and sequences may make much more sense or appear more regular in hex than in inverted decimal. The screen locations for color graphics, hi-res graphics and the alphanumeric characters are much simpler to understand and much more "logical" in hex notation than in inverted decimal. So switching the coding can sometimes give you insight into what is happening, or point to a newer or better way of doing things.

Finally, if you are studying someone else's BASIC program and see a mysterious CALL 1002, how can you tell what it means? Only with a back and forth conversion can you find out

```
CALL (−32767−1)
PRINT PEEK (−32767−1)
```

Table 1. Location hex 8000 is defined as inverted decimal −32768. Here are two ways to get there without an error message.

Don Lancaster
Synergetics

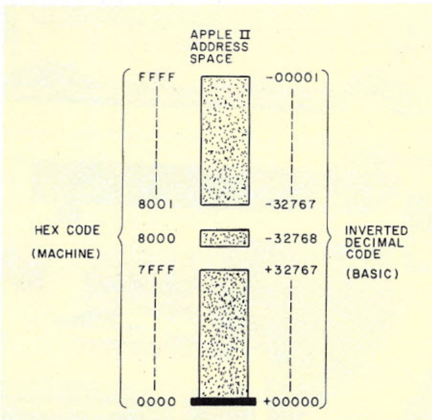


Fig. 1. Two ways of mapping Apple's address space—hex for machine language and inverted decimal for BASIC.

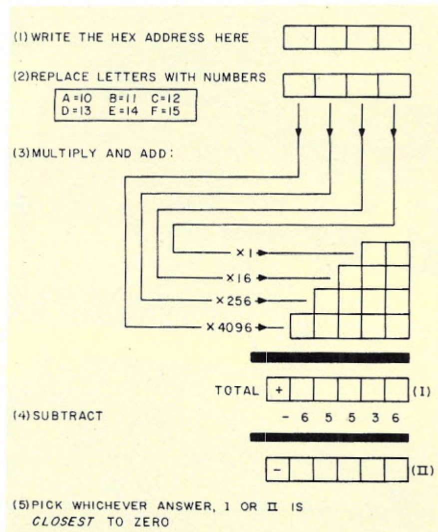


Fig. 2. Worksheet to convert hex locations to inverted decimal locations.

what the program is really up to.

For some reason, people avoid trying to convert between these two codes. Sometimes you can cheat by finding a code equivalent that may be nearby, and then counting down or up in the right direction. But, that's no fun, particularly if you go the wrong way and bomb the machine. And this avoids learning to do things the right way in the first place.

You can also generate a list and look up all the values in either direction. But this takes 65536 double entries and will produce a bunch of computer printouts. There is something better.

Scratchpad Conversions

You can look at four ways of getting from hex to inverted decimal and back again. The worksheet methods let you use a calculator and a sheet of paper. This is convenient if your Apple isn't up, or if you don't want to change what is in it, or if you just have a single location or two to convert.

Fig. 2 shows a worksheet to convert hex locations to inverted decimal locations. You can put another pad of paper along the right margin of the worksheet and work in parallel, using the same worksheet over and over again.

You use this worksheet by writing the hex address at the top and then replacing the letters with their decimal 10 through 15 equivalents. You then go through the multiply-and-add song-and-dance shown, scaling the digits by 1, 16, 256 and 4096. Total I in the figure is a direct decimal conversion of the hex address space. In other machines, this may be all you need.

But, for the Apple, you have to invert the decimal code for everything in the top half of memory. To do this, you subtract 65536 to get a new answer, II. You then pick answer I or II, the one nearest zero, as the correct conversion. Addresses 0000 to 7FFF will convert directly, while 8001 through FFFF will convert to the down-counting negative decimal values.

Fig. 3 shows two sample calculations. Here you prove that the Apple memory location "turn on the hi-res graphics" hex C057 is decimal −16297 (Fig. 3a). A second example

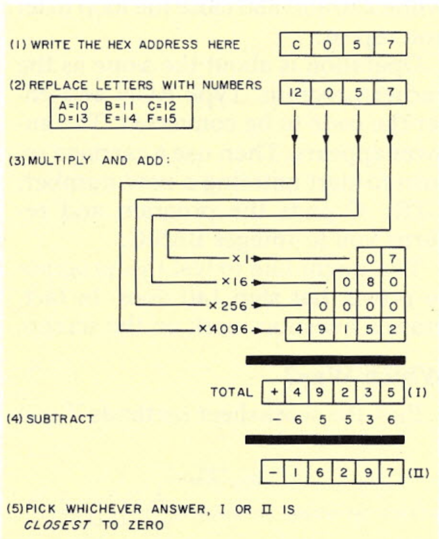


Fig. 3a. Two examples converting hex to inverted decimal.

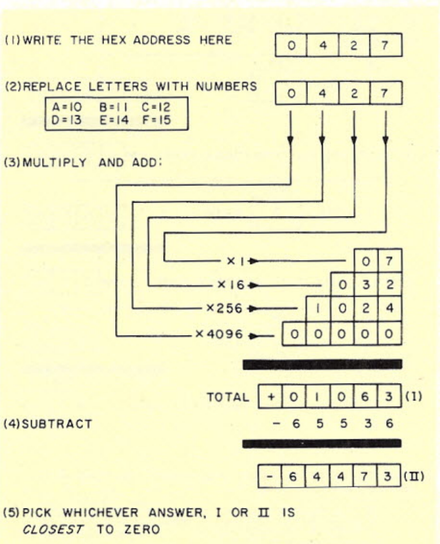


Fig. 3b. (Second example.)

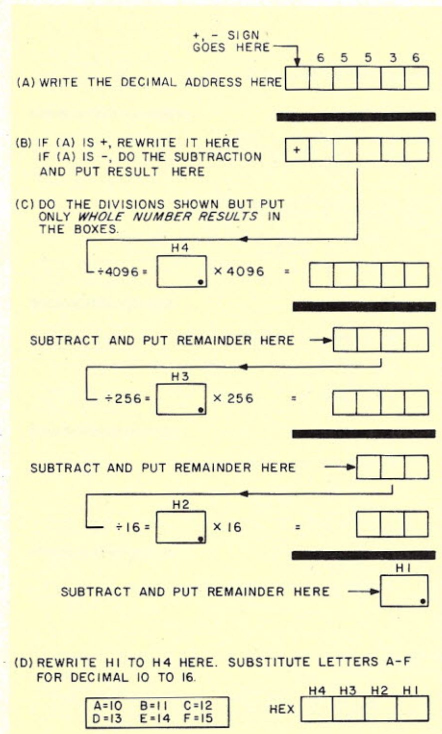


Fig. 4. Worksheet to convert decimal locations to hex locations.

✓See List of Advertisers on page 210

(Fig. 3b) shows that the upper right-hand character on the screen is both hex 0427 and decimal 1063.

Fig. 4 gives you a new worksheet to go the other way—from inverted decimal to hex. Here, you write the decimal address at the top. If the decimal address is positive, you do nothing to it. If the decimal address is negative, you subtract the address from 65536. Either way, you get a positive number from 0 to 32767.

You then go through a funny game called a math iteration. You try dividing by 4096 and write only the whole number result (the part of the answer to the left of the decimal point on your calculator) in box H4. Now, you multiply back only the whole number part of H4 and subtract to get a remainder.

Repeat this process for 256, 16 and finally 1. Each time, use only whole numbers. This gives you your raw hex digits. Finally, put all your H1 to H4 digits in order and replace the numbers above 9 with their A-F equivalents.

Two examples in Fig. 5 show the Apple locations that toggle the speaker and switch the screen from normal to inverse mode.

Conversion Programs

Listing 1 shows you a program to convert from hex to inverted decimal code. Steps 100–130 do some housework for you, clearing the screen and asking you to input the hex address. This address is read as a string (remember those letters!) called HEX$. You then read each piece of the string in steps 140–170. If you have a letter, step 170 changes the letter to its numeric 10–15 equal.

In step 190, test to see if you are working on the top half of memory. If you are, subroutine 400–420 complements the hex code for you and decides to print a minus sign in front of your final answer. If you are on the bottom half of memory, skip this step.

Then calculate the decimal equivalent in step 200 and print the answer in 210. Steps 220 and 230 do some formatting and repeating.

You can test this program with any of the previous examples. After your first answer, hit the carriage return to enter your next conversion. Do this till all the numbers you need are converted.

The reverse program trip of inverted decimal to hex is shown in Listing 2. It is somewhat longer and takes a fancier string. Again, start with housework in steps 100–130. Step 140 erases a question mark that messes up the display and starts you off assuming a lower half conversion. If you are working on the top half of the address space, step 150 goes to subroutine 400–410 to do repair work.

Step 160 sets up the string that will automatically convert numbers 10 to 15 to their hex letters. Lines 190 through 220 execute the conversion iteration; step 240 adds 8 to the leftmost hex digit if you started out with a negative inverted decimal number. Lines 250–260 print out your answer, while 270 and 280 close the loop to let you repeat.

Operation is about the same as the earlier program. Type RUN and enter the code to be converted. The answer appears. Then use a carriage return to start entering a new number. CTRL C ends the program and returns you to Integer BASIC.

You might like to test this program to prove that step 140 does, in fact, erase a question mark on the screen.

Which Way?

Pick the worksheet methods if you

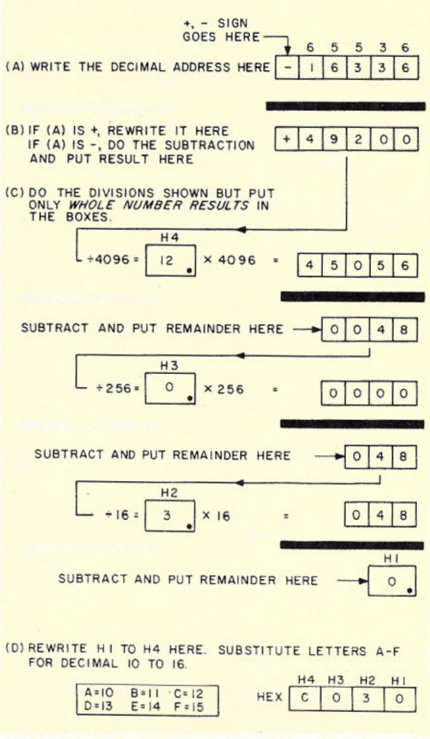


Fig. 5a. Two examples converting inverted decimal to hex.

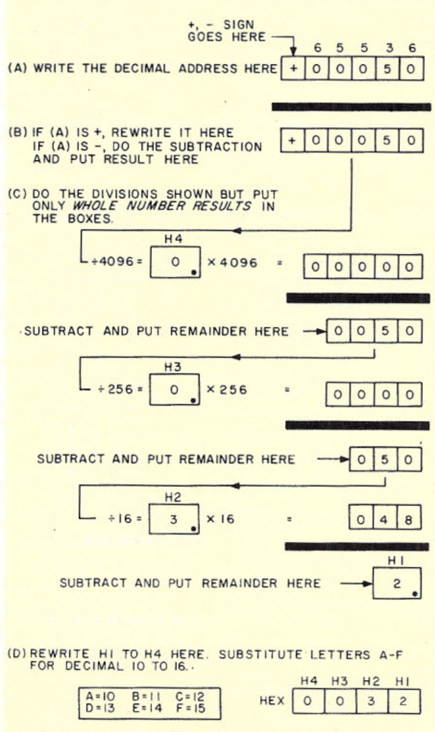


Fig. 5b. (Second example.)

have only a couple of conversions, or are away from your computer. Use the programs for everything else. You might like to combine both programs into one that goes either way. You can also use trial and error to let one program do the backwards trip for you.

The Hex 8000 Problem

Location hex 8000 looks confusing. The Apple people call this decimal −32768, but, if you try getting directly at 8000, you will probably get "> 32767 ERR". Fortunately, location 8000 rarely needs code conversion.

But there is a way. It is sneaky, but it works. Apple allows a value of −32768 *inside* the machine. It just doesn't let you put this number into the machine or take it back out any direct way.

Table 1 shows ways to get at this location. You have to give the Apple something to do that will make it internally compute a value of −32768. For instance, PEEK (−32767−1) will also work without an error message. Should you have some machine-language routine at 8000, CALL (−32767 −1) will also work without an error message.

So, don't be afraid of those funny number games involved in getting from hex to inverted decimal and back again. These codes are simple and easy to use. The two codes and their conversion are versatile tools when you learn to convert them quickly. ■

```
100 REM * APPLE HEX TO DECIMAL
110 DIM HEX$(4),A(4)
120 CALL -936: VTAB 6: TAB 10
130 INPUT "HEX ADDRESS ",HEX$
140 FOR N=1 TO 4
150 A(N)=ASC(HEX$(N,N))
160 A(N)=A(N)-176
170 IF A(N)>15 THEN A(N)=A(N)-7
180 NEXT N:HEX$=" "
190 IF A(1)>7 THEN GOSUB 400
200 DECIMAL=A(1)*4096+A(2)*256+A(3)*16+A(4)
210 PRINT : TAB 10: PRINT "IS DECIMAL ";: PRINT HEX$;: PRINT DECIMAL
220 PRINT : PRINT : PRINT : PRINT : PRINT
230 INPUT HEX$: GOTO 120
400 FOR N=1 TO 4:A(N)=15-A(N): NEXT N
410 A(4)=A(4)+1
420 HEX$="-": RETURN
```

Listing 1. Program for hex to inverted decimal conversion.

```
100 REM : APPLE DECIMAL TO HEX
110 DIM HEX$(16),A(4)
120 CALL -936: VTAB 6: TAB 10
130 INPUT "DECIMAL ADDRESS ",DECIMAL
140 POKE 1689,160:FIX=0
150 IF SGN (DECIMAL)<0 THEN GOSUB 400
160 HEX$="0123456789ABCDEF"
170 A(1)=DECIMAL/4096+1
180 R=DECIMAL MOD 4096
190 A(2)=R/256+1
200 R=R MOD 256
210 A(3)=R/16+1
220 A(4)=R MOD 16+1
230 PRINT : TAB 13: PRINT "IS HEX ";
240 IF FIX=1 THEN A(1)=A(1)+8
250 FOR N=1 TO 4
260 PRINT HEX$(A(N),A(N));
270 NEXT N: PRINT : PRINT : PRINT
280 INPUT HEX$: GOTO 120
400 DECIMAL=32767+DECIMAL+1:FIX=1
410 RETURN
```

Listing 2. Program for inverted decimal to hex conversion.

↙See List of Advertisers on page 210

The author reveals a useful key to printing characters on the computer's screen.

Poking the Apple's Screen

By Patrick C. Moyer

The BASIC print command does not always provide the flexibility that a programmer needs. One alternative is a technique that uses the POKE command to display information on the screen.

This technique requires you to access and modify the computer's video display or screen buffer. The buffer is that portion of the computer's memory which holds a copy of all the information currently on display. Modifying this area lets you print any character at any location on the screen without affecting any other character that is on the same line.

This article describes the technique as applied to the Apple II computer. Therefore, it is not applicable to graphics, but is relatively easy to apply to text display.

Screen Buffer

The first step is to locate the screen buffer in memory. This is done using a memory map, which, for the Apple II, can be found on page 126 of the Applesoft reference manual. It indicates that the screen memory is located at hexadecimal addresses 400–7FF. Since this area of memory will be accessed using the BASIC POKE command, you must convert these addresses to decimal. Thus, the decimal numbers 1024–2047 will be used in the POKEs. A quick subtraction reveals that there are 1024 memory locations (40 characters on 24 lines).

The Apple II screen buffer is not organized as a continuous memory area but as eight 128-byte segments (Table 1). Each segment contains 120 bytes of memory which correspond to 120 print positions on the Apple screen, followed by eight additional bytes. Each segment contains three 40-character (bytes) lines of data. These lines do not appear consecutively on the screen.

The screen is broken into three divisions (Table 2). Each segment of memory contains one line from each division. The first segment contains memory locations for lines 1, 9, 17; the second segment lines 2, 10, 18; and so on. The first 40 bytes correspond to locations from left to right on the first line (addresses 1024–1063), the next 40 are for print positions on the eighth line and the last 40 bytes for characters on line 17. To print a character on the screen you must know the segment the line appears in and the tab position to find the exact address to poke.

To simplify this process use Table 2. This worksheet has the address for the beginning and end of each line in the left and right margins, and a scale from 0–39 across the top and bottom which indicates each print position. By adding or subtracting this amount to the margin address, any position on the Apple screen can be located.

For example, if a character is to be placed at line 10 and tab position 10, the procedure is:

1. Look down the left side of the worksheet and find line 10 and the associated address: 1192.

2. Count over 10 spaces and read the scale: 9.

3. Add the address to the scale number to obtain the location to be poked: 1192 + 9 = 1201

This is the procedure to locate any positions on the Apple screen. This position is the first parameter for the POKE command; i.e., the address.

ASCII Codes

But what values should be poked? The second parameter of a POKE is the value that is to be stored in the address. For this methodology the ASCII equivalent values of the letters and numbers are used. The Apple provides three ways to display the characters: normal, reverse and flashing. All three are accessible using the POKE command.

On pages 138–139 of the Applesoft manual is a list of the ASCII codes for

Segments		Eight 128-byte segments 40 character lines			
	1024				
1.	40 Bytes—Line 1	40 Bytes—Line 9	40 Bytes—Line 17	8 Bytes	
2.	40 Bytes—Line 2	40 Bytes—Line 10	40 Bytes—Line 18	8 Bytes	
3.	40 Bytes—Line 3	40 Bytes—Line 11	40 Bytes—Line 19	8 Bytes	
4.	40 Bytes—Line 4	40 Bytes—Line 12	40 Bytes—Line 20	8 Bytes	
5.	40 Bytes—Line 5	40 Bytes—Line 13	40 Bytes—Line 21	8 Bytes	
6.	40 Bytes—Line 6	40 Bytes—Line 14	40 Bytes—Line 22	8 Bytes	
7.	40 Bytes—Line 7	40 Bytes—Line 15	40 Bytes—Line 23	8 Bytes	
8.	40 Bytes—Line 8	40 Bytes—Line 16	40 Bytes—Line 24	8 Bytes	
					2047

Table 1. Map of Apple screen buffer.

Patrick C. Moyer is the director of research and development at TYC Software, 40 Stuyvesant Manor, Geneseo, NY 14454.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39

1.	DIVISION 1	1024	1063	DIVISION 1	1.
2.		1152	1191		2.
3.		1280	1319		3.
4.		1408	1447		4.
5.		1536	1575		5.
6.		1664	1703		6.
7.		1792	1831		7.
8.		1920	1959		8.
9.	DIVISION 2	1064	1103	DIVISION 2	9.
10.		1192	1231		10.
11.		1320	1359		11.
12.		1448	1487		12.
13.		1576	1615		13.
14.		1704	1743		14.
15.		1832	1871		15.
16.		1960	1999		16.
17.	DIVISION 3	1104	1143	DIVISION 3	17.
18.		1232	1271		18.
19.		1360	1399		19.
20.		1488	1527		20.
21.		1616	1655		21.
22.		1744	1783		22.
23.		1872	1911		23.
24.		2000	2039		24.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39

Table 2. Worksheet of the Apple screen poking.

letters, numbers and characters. These are the codes that are produced by the BASIC function ASC (p. 60 of the Applesoft manual). These codes, when poked onto the screen, do not produce the normal version of the characters but the flashing version. Thus, if 1024, 65 is poked, the Apple will print the letter A in the upper left-hand corner, in the flashing mode. (The decimal version of the ASCII code for A is 65.)

To access the other versions of the characters, another set of codes is used. The complete list of codes needed to access all three modes using POKE are listed in Table 3. The codes for each mode are numerically related. Adding 64 to the reverse code gives the flashing code; adding 64 to the flashing code gives the normal character code.

It may be more useful to look at this in terms of the ASC function. For example, POKE 1024, ASC("A") displays a flashing mode A on the screen (the same as POKE 1024,65); POKE 1024,ASC("A")−64 produces a reverse A; and POKE 1024,ASC("A")+128 displays a normal mode A. Using this relationship, the ASC function can selectively translate character strings and print them on the screen in a variety of modes on the same line. Another application of this technique is to subtract or add to a currently displayed character, thereby changing its mode.

This technique of poking characters onto the video screen has many possibilities, and frees the programmer from some of the limitations and nuisances of the standard print command. One caution: indiscriminate poking can cause the computer to crash, resulting in the loss of the program. So be careful to check that the addresses poked are valid for the screen before running them. A major drawback of this technique is that only one character can be printed per POKE. This entails some additional programming, but the flexibility is usually worth the effort.■

	Reverse	Flashing	Normal		Reverse	Flashing	Normal
@	0	64	128	Space	32	96	160
A	1	65	129	!	33	97	161
B	2	66	130	"	34	98	162
C	3	67	131	#	35	99	163
D	4	68	132	$	36	100	164
E	5	69	133	%	37	101	165
F	6	70	134	&	38	102	166
G	7	71	135	'	39	103	167
H	8	72	136	(	40	104	168
I	9	73	137	)	41	105	169
J	10	74	138	*	42	106	170
K	11	75	139	+	43	107	171
L	12	76	140	,	44	108	172
M	13	77	141	-	45	109	173
N	14	78	142	.	46	110	174
O	15	79	143	/	47	111	175
P	16	80	144	0	48	112	176
Q	17	81	145	1	49	113	177
R	18	82	146	2	50	114	178
S	19	83	147	3	51	115	179
T	20	84	148	4	52	116	180
U	21	85	149	5	53	117	181
V	22	86	150	6	54	118	182
W	23	87	151	7	55	119	183
X	24	88	152	8	56	120	184
Y	25	89	153	9	57	121	185
Z	26	90	154	:	58	122	186
[	27	91	155	;	59	123	187
\	28	92	156	<	60	124	188
]	29	93	157	=	61	125	189
^	30	94	158	>	62	126	190
—	31	95	159	?	63	127	191

Table 3. ASCII codes for poking.

Get the jump on the rest of the field with this simple-to-use ASCII character generator.

Build a "Quick Fox" Terminal Tester

By T. K. Davies

One of the recurring requirements for the person who installs or repairs computer equipment is a black box to plug in and turn on which continuously sends ASCII characters. Whether he is building a new serial I/O board, troubleshooting a video terminal or trying to adjust a teletypewriter, it is hard to adjust or measure in one place and, with the free hand, punch keys on a keyboard.

This black box, which will free his hands, is a "quick brown fox" generator. Historically, these were large mechanical devices which had a character wheel with "the quick. . . fox. . . ." coded on it, a contact-type reader and a serializer. Such a device is hardly reasonable for today's computer work, so the following "quick-fox" unit was developed (Photo 1).

Before the project was started, I drew up a "wish list" of desirable features:

1. The unit must generate all printable characters.
2. It should be useful for both RS-232 and 20 mA loops.
3. It should have selectable baud rates.
4. The unit should be portable, battery-powered and easy on the batteries.
5. It should be easy to operate and connect to commercial terminals.
6. It should be inexpensive and easy to build from readily available parts.

ROM Character Generators

Designing the Q-Fox to meet these needs required that many alternatives be examined. Referring to the block diagram, Fig. 1, you will see that the first design consideration was the character generator. The most obvious choice for generating all the characters would be a commercial ROM with all the ASCII characters precoded.

An examination of several manufacturers turned up National's MM5220DF quick-brown-fox generator ROM, but it has several drawbacks. It does not generate all the ASCII characters; it is a PMOS chip and requires level shifters for use with CMOS ICs; and it needs several supplies and is consequently a power hog. So this was ruled out.

My next thought was to use a custom PROM and program it to do the job. Since I need only a little over 64 characters, it is hardly a job for the more common PROMs such as a 2708 or 1702. Also, neither of these chips is material for a battery-supplied test set. There are CMOS PROMs, but they are expensive and not readily programmable by most available PROM burners.

Up Counter Character Generator

After looking at the character set that was needed (all ASCII codes from octal 040 to 140 if uppercase only, or to octal 200 if both uppercase and lowercase are required), I saw that the sequential set could be generated by an up counter. Just preload with the first character (space 040) and run up to the last character ("←"

Photo 1. Q-Fox on background of sample printout.

T. K. Davies is a chemistry professor at the University of Victoria, Victoria, B.C., Canada V8W 2Y2.

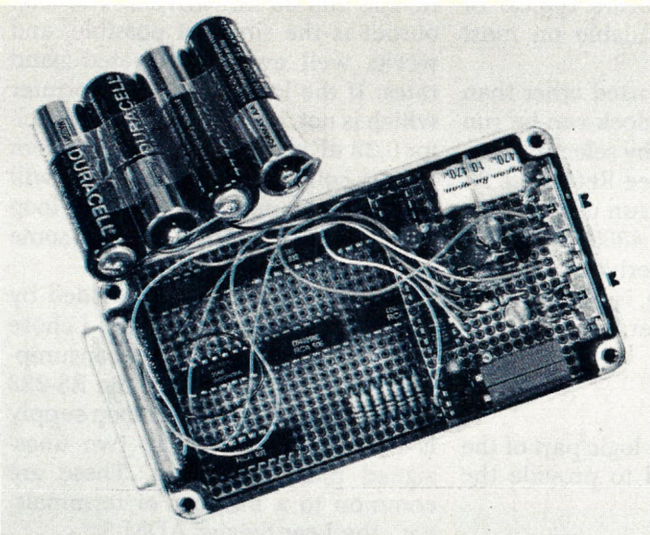


Photo 2. Interior view with batteries removed. But still assembled.

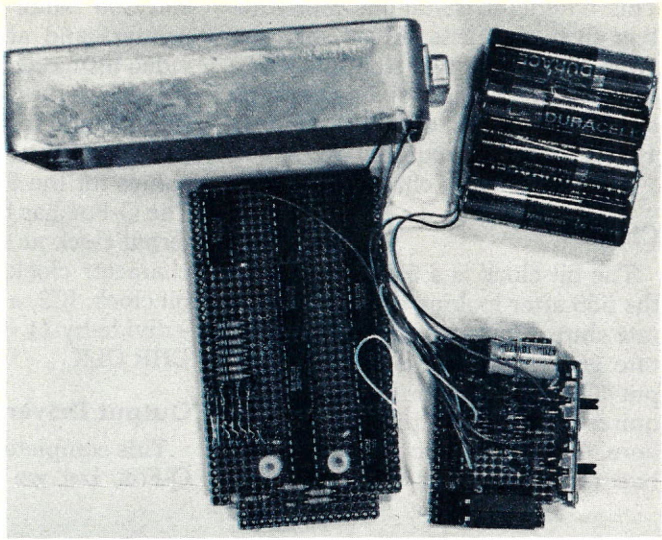


Photo 3. All boards removed from case.

137 or DEL 177), reset and repeat the sequence. This part of the circuit is done by IC4 and IC5, two cascaded 4029s.

If nothing more were required to generate a string of characters on a terminal, the generator would be trivial. But if you run this device on a teleprinter, it will print a line and then pile up all successive lines in the last column. What is needed is some means of generating, at a minimum, a carriage return and line feed at the end of the sequence. If the unit is to be used with line printers, there must also be provision for adding several null or fill characters after the CR and LF, to allow the carriage to completely return and stabilize in column one.

Add Ten Characters

This is where the block labeled ADD TEN CHRS comes into play. The circuit is composed of IC8, IC13 and parts of IC7 and IC12. It operates as follows. The section called 140 DECODE (IC7) senses that the last character has been generated (actually, it is one more than the last character, but it simplifies decoding and helps the parts count).

The decoded pulse is inverted in a second gate of IC7 called POWER ON RESET. In conditions other than just after turn on, this is only an inverter; however, POWER ON RESET and the inverter make sure that the line of characters starts at 040 (a space), and not with some random character. The inverted DECODE 140 pulse sets the RS flip-flop IC8. This, in turn, puts D of the second flip-flop high, and on the next clock pulse Q will go high. With Q high, the associated AND gate (IC7a and an inverter) will pass clock pulses to the IC13, the ten-character scanner.

IC13 acts like a bucket brigade, in that a high or one is passed sequentially from 0 to 9. During the scan the diode matrix elements are connected one after another. When the high reaches 9, a pulse is sent to the reset line of IC8; this causes Q to go low and D of the next flip-flop to go low, which stops the clock pulses from reaching IC13. Thus, only ten pulses are generated until the DECODE 140 gate signals to send another ten. If, for some reason, a shorter sequence than ten is required, just take off the reset line to IC8 at an earlier point.

Custom Characters

These custom characters are generated by jamming ten characters into the counter, which in this mode is just a buffer into the parallel-to-serial converter. A diode matrix is connected across the load input of the two 4029 counters. The load inputs are pulled down (low or zero load), but during the add-ten-character sequence they can be pulled high (one) by diodes of the programming array.

The diagram for the array (see Fig. 2) shows an N.C. for the first character; in effect, this is a null. The settling time of the add-ten circuit requires that this be a null. Next are the CR, LF and two fill characters. If you are just using the Q-Fox for video terminals, these are not required. The next four characters are vanity characters, in our case UVIC, and the last character is a space, which is also required since it is the startpoint for the up counters. To custom-program the optional characters, look up the ASCII equivalents and add the appropriate diodes to the matrix. The octal values for each input are identified on the schematic (Fig. 2).

Serializer

The next consideration in building the Q-Fox was the serializer. My first approach was a CMOS UART. Although this worked well and used low power, it was not used in the final version because it was costly, large and had many redundant features for this application. The serializer comprised a 4021 and a 4013.

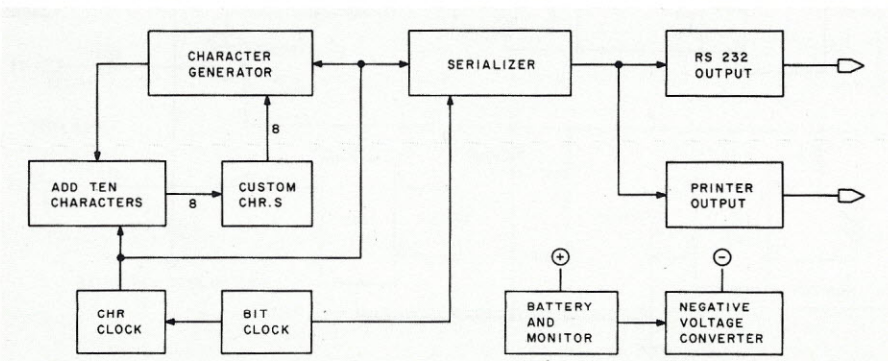


Fig. 1. Block diagram for Q-Fox computer terminal tester.

This parallel-in serial-out configuration gives eight bits of data, and provides a start and two stop bits. For a detailed description of its operation see the *CMOS Cookbook* (Reference 1). The serializer requires two clocks —a bit clock and a character clock.

Clocks

The bit clock is a 555-IC1. I chose the 555 after looking at several baud rate chips. One drawback of the baud rate generators is the 16-times output. This output is for UARTs and requires a divide-by-16 to be useful here. The schematic (Fig. 2) shows two rates and gives values for 110 and 300 baud; these are typical of printers and are available on most video terminals.

If the unit is to be used other than for printers, the bit clock can be run at a higher baud rate by selecting new values for the 555 (see Reference 2). The Q-Fox has been run using an external clock at up to 4800 baud. The character clock is derived from the bit clock. IC2, a 4029, is connected as a divide-by-11 counter to provide the CHR CLK.

Output Drivers

This completes the logic part of the Q-Fox, but we need to provide the RS-232 and 20 mA drivers. The loop output is the simplest possible, and works well even at elevated baud rates. If the loop is driving a printer which is not despiked, a small capacitor 0.22 uF or so could be added from the Q1 collector to ground. This will reduce the upper rate for the loop supply but will aid in driving some printers.

The RS-232 output is provided by an operational amplifier IC9. I chose a 1458 for its low power consumption. The output pins for the RS-232 are standard, but, since a loop supply is not an RS-232 signal, two unassigned pins were used. These are common to a number of terminals, e.g., the Lear Seigler ADM-3.

Both of the output drivers require bipolar supplies. RS-232 switching can be accomplished by 3 to 4 V, but the loop supply needs 5–6 V. The prototype was built with two batteries and two 9 V transistor batteries. This worked well, except that the batteries took up as much room as the circuit, and, at 9 V, the CMOS drew nearly twice the current. My next choice was a new chip from Intersil. The ICL 7660 is a negative voltage converter. Since this is a new chip it is worth a few lines of comment.

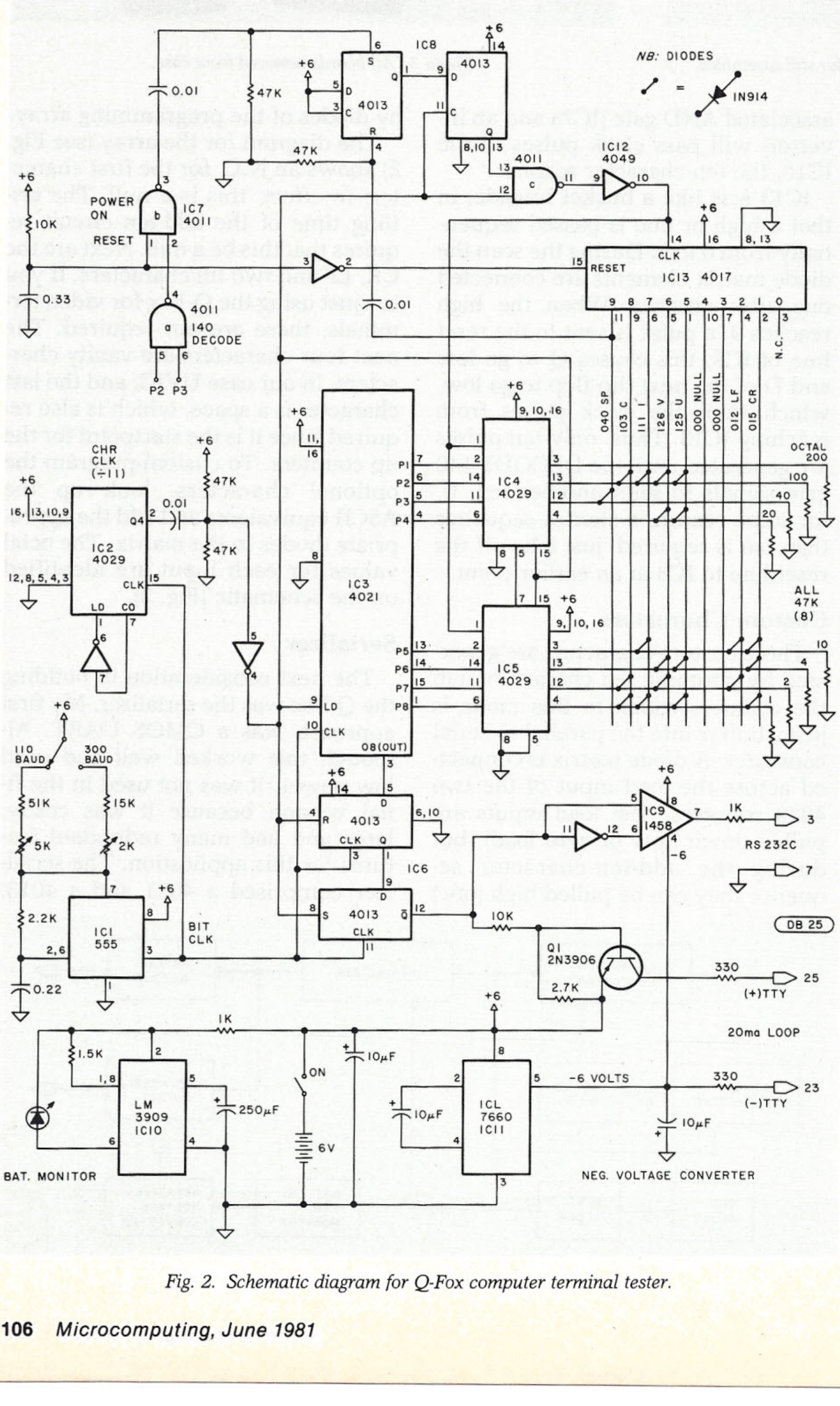


Fig. 2. Schematic diagram for Q-Fox computer terminal tester.

The ICL 7660

This is a battery (or supply) inverter chip. With the addition of three or four components, supplies from −1.5 to −10 V can be generated. The chip contains all the parts of a voltage doubler. By adding the external capacitors, the unit is complete.

An RC oscillator and several MOS switches (which act like diodes on a conventional doubler) convert and invert the positive supply. The voltage is then regulated. The circuit works with high efficiency, typically 98 percent, in most applications. It can be used for dynamic RAM or processors which require negative supplies, or, as in our case, an operational amplifier and level translator. I found it needed a capacitor across the main supply rail to stop the spikes from the inverter from glitching the main logic.

Battery Monitor

The last circuit required is a battery monitor, which may appear to be a frill but is necessary to ensure that the battery is in good condition. The RS-232 and the loop supply do have lower limits for proper operation. Although the main logic board will run

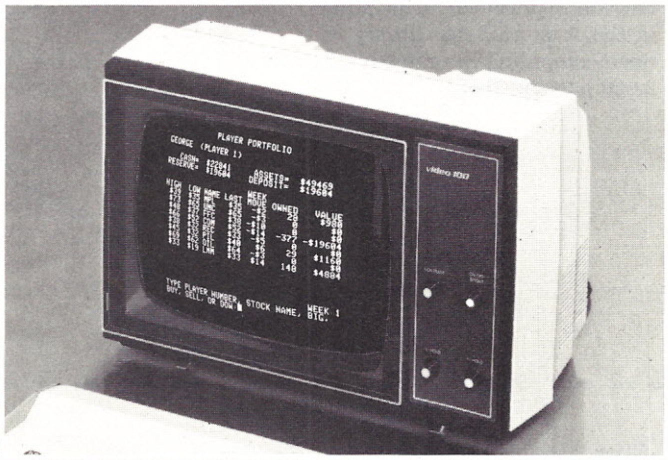
AMDEK

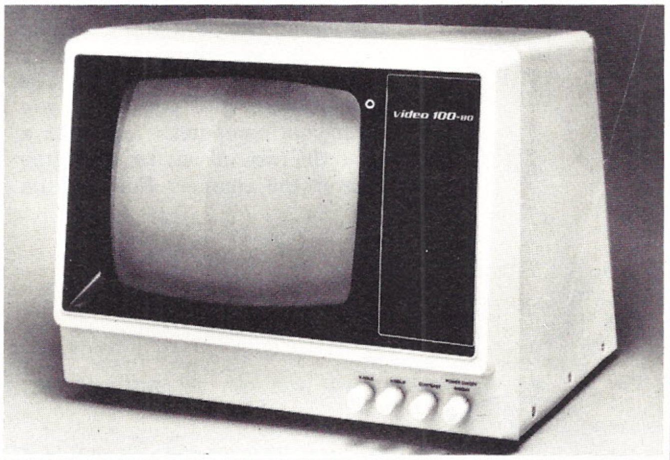
Professional Color & B/W Monitors

NEW 13″ COLOR

The Amdek 13″ Color Video Monitor is ideal for all personal and business computing applications. Bright jitter-free text allows viewing over extended periods without causing eye fatigue, especially in word processing applications. The low resolution display provides 40 characters wide by 24 characters deep with 260 horizontal lines and 300 vertical lines.

This compact, lightweight monitor (only 25 lbs.) has a molded-in carrying handle, making it easy to move. Its industrial grade construction and shatter resistant case provide reliable service in both office and industrial environments. All front mounted controls make it easily adjustable at a glance without the loss of a single keystroke.





VIDEO 100 12″ B/W MONITOR

This highly reliable 12″ black and white monitor features a 12MHz band width and 80 character by 24 line display. Plug-in compatibility with Apple, Atari, Radio Shack, O.S.I., Micro-Term and Exidy make this the perfect text display for almost any system.

VIDEO 100-80 12″ B/W MONITOR

The model 80 features an industrial grade metal cabinet with built-in disk mounting capability and space for an 11″ x 14″ PC board for custom designed electronics. Front mounted controls include power, contrast, horizontal hold, vertical hold, and brightness.

Amdek Corp. • 2420 E. Oakton Street, Suite "E" • Arlington Heights, IL 60005 • (312) 364-1180 TLX: 25-4786

166

Amdek Corp. the new name for **Leedex . . .** Expanding for the future, growing with success.

to well below 3 V, the output drivers would be marginal well before this point. The battery monitor will register if the battery is above 5 V; this means that when the LED is flashing the unit is working properly.

Alkaline batteries were used to ensure maximum running time. Regular AA cells will work just as well, but for a shorter time. The drain is 10 mA for an RS-232 unit under test and 16 mA (average) for a loop supply driven printer. The better batteries should provide months of normal operation.

Construction

The unit was constructed in a small alloy box (2.5×4.5×1.25 inches).

The generator is constructed on two boards. The main logic board is full-sized (see Photo 4) and comprises all but the battery monitor, RS-232 and TTY drivers and the negative voltage convertor, which are on a partial board (see Photo 5). The batteries are solder-connected, wrapped in a thin piece of foam and set in at the end of the small board.

The power switch and the baud rate switch protrude through the case at one end, and the RS-232 (DB25) connector protrudes through the other end; a hole for the LED is on the top of the box. The boards are cut from a Vector 4112-4DP universal board. If you want to duplicate the unit exactly, note that the bottom section is three rows wide (running end to end) and the small board is two rows (running side to side). The unit seems to be insensitive to parts layout, so any reasonable configuration would probably be all right.

How to Use It

A few comments on the use of the Q-Fox would be worthwhile. Connection to a Teletype, such as the model 33, is straightforward. Pin 25 of the DB25 plug is the positive of the loop supply, and pin 23, the negative. Connection for RS-232 is on pin 3 for signal and pin 7 for ground. Note that some terminals used with modems and line drivers require jumpers to enable the RS-232 line—the easiest way to check is to look at the cable ends of the particular terminal and see if jumpers are present on the cable plug. The unit has been used on Lear Seigler, Heath and Westinghouse terminals and model 33 and 35 Teletypes.

If you are plugging the unit directly into the terminal, a male plug is required; if using it as a computer end of the RS-232 cable, use a female socket, DB25 connector. If using it for a variety of applications, you may need to make some DB25 female-to-female adapters, or male-to-female if the socket is recessed on the terminal. Determine your requirement before you install the DB25 connector.■

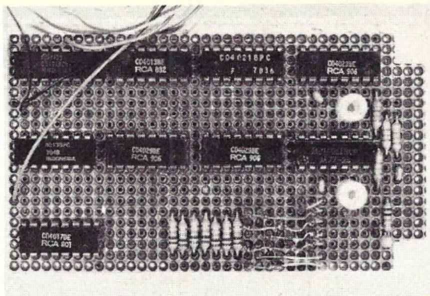


Photo 4. Closeup of the main logic board (note stand offs to support upper small board). Diode matrix is in the bottom right. RC components are on the end of the board.

References

Lancaster, D., *CMOS Cookbook*, Sams & Co., New York, p. 317.

Lancaster, D., *TTL Cookbook*, Sams & Co., New York, p. 171.

Parts List

IC1 LM 555
IC2 4029
IC3 4021
IC4 4029
IC5 4029
IC6 4013
IC7 4011
IC8 4013
IC9 MC 1458
IC10 LM 3909
IC11 ICL 7660
IC12 4049
IC13 4017

Q1 2N3906
Diodes 1N914

1—FLV-117 LED

3—10 uF/10 V tant.
1—0.22 uF/10 V tant.
3—.01 uF Cer.
1—250 uF/10 V electrolytic
1—0.33 uF/10 V tant.

12—47k 1/4W resistors
2—10k 1/4W resistors
2—1k 1/4W resistors
2—330k 1/4W resistors
1—1.5k 1/4W resistors
1—2.7k 1/4W resistors
1—2.2k 1/4W resistors

1—15k 1/4W resistor
1—51k 1/4W resistor

1—5k trimpots
1—2k trimpots

1—DB25 (RS-232 connector).
1—Vector 4112-4DB PC board.
2—SPDT slide switches.
1—Hammond 1590B utility box.

4—Mallory MN1500 1.5 AA cells.

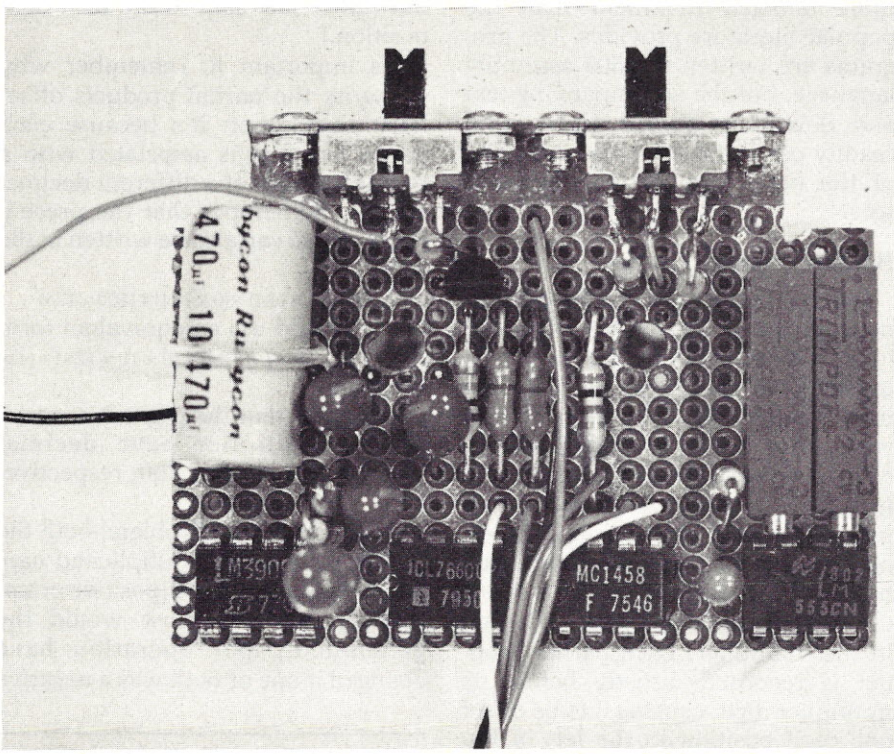


Photo 5. Closeup of output drivers, battery monitor, inverter and bit clock.

No more tedious times tables to recite; try binary multiplication in 6502 assembly language.

Multiplying by 1's and 0's

By Leo J. Scanlon

Multiplication. It's a subject that gave most of us untold misery in elementary school. But now, having memorized all of those cursed multiplication tables, we can confidently multiply anything by anything. Right?

Yes, right, as long as the anythings are *decimal* numbers. Unfortunately, if you want to write an assembly language program to perform a multiplication on your Apple, AIM 65, TRS-80 or other microcomputer, you're back at square one, because (a) the numbers being multiplied are *binary* values and (b) none of the popular eight-bit microprocessors has a multiply instruction.

That's the bad news. The good news is that multiplication techniques are covered in virtually every tutorial and textbook on assembly language programming. Regardless of which book you pick up, you'll learn how it's possible to multiply two binary values, using the add and shift instructions of the microprocessor. The discussions of multiplication vary in quality—depending on the author's inclination and the software or hardware orientation of the book—but typically these books give only a summary of the fundamental principles and an example or two (usually, just an eight-bit by eight-bit unsigned multiply). From there on, readers are left to their own devices.

A cop-out? Maybe. But most authors (and, apparently, their publishers) feel that their book must cover so many topics that they can't afford to devote much page space to a "simple" task like multiplication.

Admittedly, a brief once-over will suit the needs of the casual reader/programmer, but what about those who want to multiply numbers that are longer than eight bits, or those who want to multiply *signed* (two's complement) numbers?

This article is intended to serve both kinds of readers: beginners who want an overview of multiplication and serious programmers who need a more detailed treatment than the popular literature provides. The programs are written in 6502 assembly language, but the accompanying text and flowcharts should make them readily convertible for use with any of the other eight-bit microprocessors.

Back to Fourth Grade

Before we discuss multiplying binary numbers, it may be instructive to review the mechanics of multiplying decimal numbers, the way we used to do it in elementary school with pencil and paper. As you will recall (in these days of calculators, it may be a bit hazy), with pencil and paper you write the multiplicand on one line and the multiplier on the line below it, and then grind out a series of multiplications—one for each digit in the multiplier. Each partial product is recorded directly below its multiplier digit, causing it to be offset one digit position to the left of the preceding partial product. When all of the partial products have been calculated, they are added to produce the final product.

For example, multiplying the number 124 by the number 103 looks like this:

124	Miltiplicand
× 103	Multiplier
372	Partial Product #1
000	Partial Product #2
124	Partial Product #3
12772	Final Product

(Of course, you don't normally write down the all-zeroes partial product, but rather just skip to the next digit position.)

It's important to remember why we write the partial products offset from each other: it's because each partial product is associated with a multiplier digit of a different decimal weight. Remember that the preceding problem can also be written in the form

$103 \times 124 = (3 \times 124) + (0 \times 124) + (100 \times 124)$

Or, we could use an equivalent form

$103 \times 124 = (3 \times 1 \times 124) + (0 \times 10 \times 124) + (1 \times 100 \times 124)$

to illustrate that the digits 3, 0 and 1 in the multiplier have decimal weights of 1, 10 and 100, respectively.

In the preceding problem, both the multiplier and the multiplicand happen to be nonnegative (positive or unsigned) numbers. How would the pencil-and-paper operation have changed if one or both was a negative

Address correspondence to Leo J. Scanlon, 23021A Village Drive, El Toro, CA 92630.

```
This subroutine multiplies an 8-bit unsigned multiplicand (MPND)
by an 8-bit unsigned multiplier (MPLR), and returns the 16-bit
product in locations PROD (low byte) and PROD+1 (high byte).

0000                  MPLR=$20
0000                  MPND=$21
0000                  PROD=$22
0000                  *=$400
0400 A9 00  MLT8U     LDA  #0       Clear product MSBY
0402 A2 08            LDX  #8       Multiplier bit count = 8
0404 46 20  NXTBT     LSR  MPLR     Get next multiplier bit
0406 90 03            BCC  ALIGN    Multiplier bit = 1?
0408 18               CLC           Yes, add multiplicand
0409 65 21            ADC  MPND      to partial product
040B 6A     ALIGN     ROR  A        Rotate product right
040C 66 22            ROR  PROD
040E CA               DEX
040F D0 F3            BNE  NXTBT    Loop until 8 bits are done
0411 85 23            STA  PROD+1   Store product MSBY
0413 60               RTS
```

Example 1. An eight-bit by eight-bit unsigned multiplication subroutine.

number? Very little, because we all know that if one number is positive and the other is negative, the product will be negative—so you tack a minus sign onto the answer. Similarly, if both multiplier and multiplicand are negative, the product will be positive —so you omit the minus sign from the answer (or put a plus sign on it, if you're a purist). With binary numbers, though, there's quite a bit of difference between multiplying unsigned numbers and multiplying signed numbers, because signed numbers are represented in two's complement form. We'll be looking at both unsigned and signed multiplication later, but for now, let's briefly discuss how binary numbers can be multiplied.

Binary Multiplicaton vs. Decimal Multiplication

Binary multiplication is much simpler than decimal multiplication, because binary multipliers consist of only the digits 0 and 1, whereas decimal multipliers can be made up of the digits 0 through 9. In binary multiplication, the partial product will always be simply a copy of the multiplicand if the multiplier digit is 1, and it will be 0 if the multiplier digit is 0.

The binary equivalent of our previous 103 × 124 example looks like this:

```
        01111100  Multiplicand (=124)
      × 01100111  Multiplier (=103)
        01111100
       01111100
      01111100
     00000000
    00000000
   01111100
  01111100
 00000000
011000111100100  Final Product (=12772)
```

This example gives a good indication of the way in which eight-bit microprocessors must perform multiplication operations. There are some important differences, however. When performing a binary multiplication by hand, you must calculate the final product by adding the partial products, column-by-column. (And if you think that's easy, try it! The carries can drive you bonkers.) The operation is similar on a computer, but instead of waiting until all of the individual partial products are calculated before deriving the final product, *computer programs update the partial product after each multiplier bit is examined.* By doing this, the final product is generated when the last bit of the multiplier has been processed.

Moreover, when multiplying by hand, each partial product is offset one digit position to the left of the preceding partial product, to account for the weight of the multiplier bit. In a computer, it is easier to shift the partial product each time it is updated, thereby aligning it to receive the contribution of the next multiplier bit. The partial product may be shifted either right or left, depending on whether your program is examining the multiplier bits from right to left (low-order to high-order) or from left to right (high-order to low-order). In this article, the multiplier will be processed right-to-left, the way you would do it by hand, so the partial product will be shifted to the right.

In summary, the following applies when multiplying binary numbers by a computer:

If the multiplier bit is a 1, add the multiplicand to the partial product, and then shift the sum one bit position to the right. If the multiplier bit is a 0, shift the current partial product one bit position to the right, with no addition.

In writing a multiplication program for your microcomputer, which instruction would you expect to use to perform the add and right-shift operations? With a 6502-based microcomputer, such as Apple II, KIM-1, SYM-1 or AIM 65, the addition will be performed with the 6502's only add instruction, add to accumulator with carry (ADC). The shifting will be performed with the shift right (LSR) or rotate right (ROR) instruction.

With this background, let's examine the software that will be needed to multiply unsigned or signed numbers.

Multiplying Unsigned Numbers

As you probably know, in an unsigned number, each data bit carries a certain binary weight, according to its position within the number. Data bits are numbered from right to left, with the rightmost bit labeled as bit 0 and the leftmost bit labeled bit 7. The bit numbering scheme has a direct correlation to the binary weights, in that bit 0 has a weight of 2^0 (decimal 1) and bit 7 has a weight of 2^7 (decimal 128). Therefore, a single byte can represent an unsigned number from decimal 0 (binary 00000000) to decimal 255 (binary 11111111).

Single-Precision Unsigned Multiplication

Certainly, the easiest place to begin is by developing a program—a subroutine, actually—to multiply two eight-bit unsigned numbers in memory. If the multiplicand and the multiplier are both eight bits long, how long will the product be? Well, we know that the worst case involves multiplying 255 by 255, which gives a product of 65,025. To hold a value of 65,025, we need 16 bits, or two bytes.

At this point, we can draw the flowchart that will serve as the blueprint for an eight-bit (or single-precision) multiplication subroutine. This flowchart must do five things:

1. Initialize a two-byte product in memory to zero, and a multiplier bit counter to eight.
2. Shift the multiplier right one bit position into carry.
3. Interrogate the state of the multiplier bit that's in carry. If carry = 1, add the multiplicand to the partial product.
4. Rotate the partial product right, into the final product location's least-

```
This subroutine multiplies a 16-bit unsigned multiplicand (MPND,
MPND+1) by a 16-bit unsigned multiplier (MPLR, MPLR+1), and
returns the 32-bit product in locations PROD (low byte), PROD+1,
PROD+2 and PROD+3 (high byte).

0000                MPLR=$20
0000                MPND=$22
0000                PROD=$24
0000                *=$400
0400 A9 00   MLT16  LDA  #0        Clear P2 and P3 of product
0402 85 26          STA  PROD+2
0404 85 27          STA  PROD+3
0406 A2 10          LDX  #16       Multiplier bit count = 16
0408 46 21   NXTBT  LSR  MPLR+1    Get next multiplier bit
040A 66 20          ROR  MPLR       into Carry
040C 90 0B          BCC  ALIGN     Multiplier bit = 1?
040E A5 26          LDA  PROD+2    Yes, fetch P2
0410 18             CLC             and add M0 to it
0411 65 22          ADC  MPND
0413 85 26          STA  PROD+2    Store new P2
0415 A5 27          LDA  PROD+3    Fetch P3
0417 65 23          ADC  MPND+1     and add M1 to it
0419 6A      ALIGN  ROR  A         Rotate product right
041A 85 27          STA  PROD+3
041C 66 26          ROR  PROD+2
041E 66 25          ROR  PROD+1
0420 66 24          ROR  PROD
0422 CA             DEX            Decrement bit count
0423 D0 E3          BNE  NXTBT     Loop until 16 bits are done
0425 60             RTS
```

Example 2. A 16-bit by 16-bit unsigned multiplication subroutine.

significant byte (LSBY).

5. Decrement the multiplier count. If it is zero, store the final product's most-significant byte (MSBY) in memory, and return; otherwise, go back to process the next multiplier bit.

The flowchart in Fig. 1 performs these five tasks.

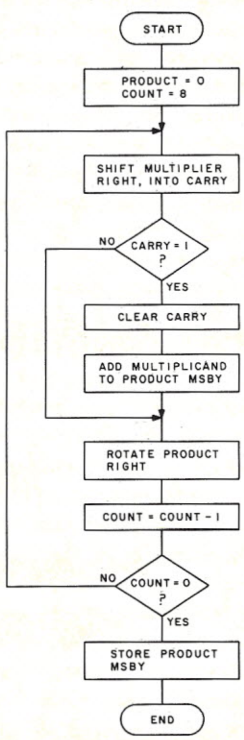


Fig. 1. An eight-bit by eight-bit unsigned multiplication algorithm.

Example 1 is a subroutine (MLT8U) that uses the flowcharted algorithm to multiply the contents of a multiplicand in memory (MPND, assigned to location $21 here) by the contents of a multiplier in memory (MPLR, assigned to location $20 here). The 16-bit product is returned in two consecutive locations, PROD and PROD + 1. The X register is used to hold the multiplier bit count.

In the MLT8U subroutine, the LSR MPLR instruction at NXTBT causes the multiplier (in memory location $20) to be shifted, one bit at a time, into carry. If the shifted multiplier bit is a one, the CLC and ADC MPND instructions add the multiplicand to the most-significant byte of the product, and the two rotate instructions at ALIGN (ROR A and ROR PROD) shift the partial product to the right, into the least-significant byte. If the shifted bit of the multiplier is a zero, BCC ALIGN bypasses the add-multiplicand operation by branching to the rotate-right sequence at ALIGN. The NXTBT loop is executed eight times, once for each bit in the multiplier.

Double-Precision Unsigned Multiplication

We've just concluded a discussion of eight-bit by eight-bit unsigned multiplication, which lets us multiply two numbers as large as 255. Unfortunately, like a three-legged race team in the Boston Marathon, single-precision multiplication is nice, but

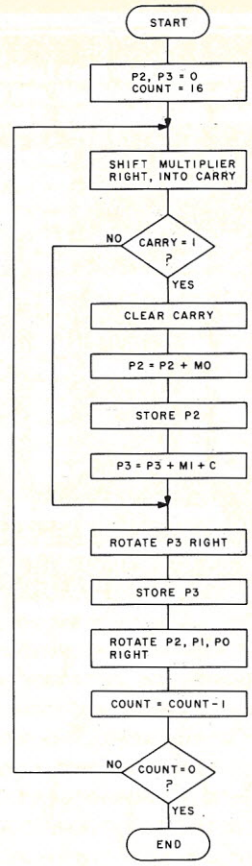


Fig. 2. A 16-bit by 16-bit unsigned multiplication algorithm.

not very practical, since many applications require numbers larger than 255 to be multiplied. Obviously, the next step is to develop a program that will multiply 16-bit, or double-precision, numbers.

Multiplying 16-bit numbers is somewhat more complex than multiplying eight-bit numbers, due to the additional memory involved, but the add-and-shift procedure still applies. With a double-precision multiplication, the multipler and multiplicand are both 16-bit values, so the product will occupy 32 bits (or four bytes) in memory.

Fig. 2 is a flowchart for a double-precision unsigned multiplication subroutine. The two-byte multiplicand is represented by the symbols M0 (low order byte) and M1 (high-order byte). The four-byte product is represented by the symbols P0 (low-order byte), P1, P2 and P3 (high-order byte). The double-precision algorithm shown in Fig. 2 operates similarly to the single-precision algorithm that has just been discussed. That is, the multiplicand is added to the high-order half of the partial product (P2 and P3, here) if carry is a 1. The result is then rotated one bit to the right,

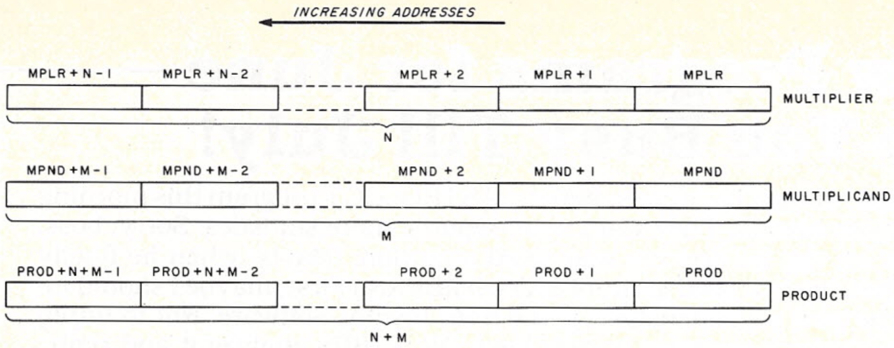


Fig. 3. Multiplier, multiplicand and product in memory.

along with the low half of the partial product.

Example 2 is a subroutine (MLT16) that uses the flowcharted algorithm (Fig. 2) to perform a double-precision unsigned multiplication. In this particular subroutine, the multiplier (MPLR) is in locations $20 and $21, the multiplicand (MPND) is in locations $22 and $23 and the 32-bit product is returned in locations $24 (low byte) through $27 (high byte). Each time a multiplier bit is a one, the MLT16 subroutine must perform two adds, to add the multiplicand to the partial product, and two stores, to update the product in memory. It must also perform four rotate operations (one for each byte in the product). Note that the updated byte P3 is returned to memory *after* the first of the four rotates.

Whither Goest the Multiplier?

If you've run either Example 1 or Example 2 on a microcomputer, you've observed that the multiplier has been cleared to zero in the course of the operation. In many applications, it makes no difference whether or not the multiplier is affected. In other applications, the multiplier must be preserved for one reason or another.

How can the multiplier be preserved? Typically, the first impulse is to preserve it by simply changing the LSR instruction at ALIGN to an ROR instruction. Well, that's a good beginning, but you must be careful to observe that for a rotate to work correctly, the sense of the carry bit must be unchanged from one rotate operation to another. Unfortunately, the add operation between labels NXTBT and ALIGN will *always* alter the carry bit!

With these two considerations in mind, we're on our way to a solution. So far, we've discovered that if you want to preserve the multiplier you must

1. Rotate, rather than shift, the multiplier, and
2. Save the state of the carry bit between rotate operations.

Will these two modifications alone preserve the multiplier? Well, almost. They don't quite do the job because the final rotate operation will leave the most-significant multiplier bit in carry, and all other less-significant multiplier bits displaced one bit position to the left. So, in addition, you must

3. Rotate the multiplier one more time, at the end of the operation.

By applying these three rules to Example 2, you may come up with a subroutine that looks like the one in Example 3, a double-precision unsigned multiplication subroutine in which the multiplier is preserved. Since the multiplier must be rotated at two different points in the program (per rules 1 and 3), the rotate instructions are given in a subroutine, called RMPLR. Rule 2 is easily satisfied by executing a push processor status (PHP) instruction after the call to RMPLR, and a complementary pull processor status (PLP) instruction just after the partial product is rotated.

Multiprecision Unsigned Multiplication

Many real-world applications involve multiplying numbers that are longer than two bytes, or have a multiplier and multiplicand of different lengths. For these reasons, it is worthwhile to wind up this discussion of unsigned multiplication by developing a subroutine that can multiply unsigned numbers of *any* length. That is, we will develop a subroutine that multiplies an M-byte multiplicand by an N-byte multiplier, and yields an (N+M)-byte product. Fig. 3 shows how these terms are stored in memory.

The overall approach is unchanged for this general case, but since the multiplier and multiplicand are variable-length values, we will have to compute parameters that were *known* in the eight-bit and 16-bit multiplications. Here, for example, the number

```
This subroutine is a modified version of the subroutine in
Example 2.  It has some additional instructions to return the
multiplier in its original form.

0000                     MPLR=$20
0000                     MPND=$22
0000                     PROD=$24
0000                     *=$400
0400 A9 00       MLT16   LDA  #0       Clear P2 and P3 of product
0402 85 26               STA  PROD+2
0404 85 27               STA  PROD+3
0406 A2 10               LDX  #16      Multiplier bit count = 16
0408 18                  CLC
0409 20 27 04    NXTBT   JSR  RMPLR    Go get next multiplier bit
040C 08                  PHP            and save it (carry)
040D 90 0B               BCC  ALIGN    Multiplier bit = 1?
040F A5 26               LDA  PROD+2   Yes, fetch P2
0411 18                  CLC            and add M0 to it
0412 65 22               ADC  MPND
0414 85 26               STA  PROD+2   Store new P2
0416 A5 27               LDA  PROD+3   Fetch P3
0418 65 23               ADC  MPND+1    and add M1 to it
041A 6A          ALIGN   ROR  A        Rotate product right
041B 85 27               STA  PROD+3
041D 66 26               ROR  PROD+2
041F 66 25               ROR  PROD+1
0421 66 24               ROR  PROD
0423 28                  PLP           Retrieve Carry from stack
0424 CA                  DEX           Decrement bit count
0425 D0 E2               BNE  NXTBT    Loop until 16 bits are done
0427 66 21       RMPLR   ROR  MPLR+1   Rotate multiplier right
0429 66 20               ROR  MPLR
042B 60                  RTS
```

Example 3. A double-precision unsigned multiplication subroutine that preserves the multiplier.

of bits in the multiplier must be determined by multiplying N by eight (with three left-shifts). The length of the product must also be computed, by adding N and M. And whenever the multiplicand needs to be added to the partial product, it will have to be added to the most-significant "M" bytes—still another required computation. These various computations are reflected in Fig. 4, the flowchart for multiprecision unsigned multiplication.

As you can see, the flowchart in Fig. 4 contains the same number of steps as the flowchart for the double-precision case (Fig. 2). That's not surprising, since the same types of operations are being performed, but how do the actual multiplication subroutines compare? That is, what programming overhead is involved in having a general-purpose subroutine rather than a length-specific subroutine? The answer is apparent by comparing Example 4, the multiprecision unsigned multiplication subroutine, with Example 3, the 16-bit multiplication subroutine. The multiprecision subroutine is *twice as long* as the double-precision subroutine!

Although Example 4 looks complicated, it isn't. Its initialization includes two additional input parameters (the multiplier length, N, and the multiplicand length, M) and two symbolic locations (to hold the product index, PINDX, and the multiplier bit count, MBIT). The subroutine itself differs very little from Example 3, except for the various computations and the use of indexes for addressing.

What lengths of numbers can be multiplied by the subroutine shown in Example 4? Well, you can see that the first seven instructions add N and M, and put the result in an eight-bit register (Y) and an eight-bit memory location (PINDX); therefore, the sum of N and M must not exceed decimal 255. The multiplier bit count ($8 \times N$) is also stored in an eight-bit memory location, so the value of N must not exceed 31. These two limitations allow us to conclude that the multiplicand length, M, must not exceed (255 − 31)

```
This subroutine multiplies two variable-length, unsigned
integers.  The multiplier is stored starting at location
MPLR, and is N bytes long.  The multiplicand is stored
starting at location MPND, and is M bytes long.  The
product will be "N + M" bytes long, and will be returned
in memory starting at location PROD.
This subroutine affects the A, X and Y registers.

0000                      N=$20        Multiplier length
0000                      M=$21        Multiplicand length
0000                      MPLR=$30     Multiplier loc.
0000                      MPND=$40     Multiplicand loc.
0000                      PROD=$50     Product loc.
0000                      PINDX *=*+1  Product index
0001                      MBIT  *=*+1  Multiplier bit count
0002                      *=$600
0600 18          MMPYU    CLC          Calculate product index
0601 A5 20                LDA  N        (N+M-1)
0603 AA                   TAX
0604 65 21                ADC  M
0606 A8                   TAY           and save it in Y
0607 88                   DEY
0608 84 00                STY  PINDX    and in PINDX
060A A9 00                LDA  #0      Clear the high M bytes
060C 99 50 00    CLRP     STA  PROD,Y   of the product
060F 88                   DEY
0610 C4 20                CPY  N
0612 B0 F8                BCS  CLRP
0614 8A                   TXA          Calculate multiplier bit count
0615 0A                   ASL  A
0616 0A                   ASL  A
0617 0A                   ASL  A
0618 85 01                STA  MBIT     and save it in MBIT
061A 20 4A 06    NXTBT    JSR  RMPLR   Get next multiplier bit
061D 08                   PHP          Save resulting Carry
061E 90 E1                BCC  ALIGN   Multiplier bit = 1?
0620 A4 20                LDY  N       Yes.  Add multiplicand to
0622 A2 00                LDX  #0       high M bytes of product
0624 18                   CLC
0625 B9 50 00    AMPND    LDA  PROD,Y
0628 75 40                ADC  MPND,X
062A 99 50 00             STA  PROD,Y
062D C8                   INY
062E E8                   INX
062F 08                   PHP          Save Carry between adds
0630 C4 00                CPY  PINDX
0632 F0 04                BEQ  PULLC
0634 90 02                BCC  PULLC
0636 B0 04                BCS  GOROT
0638 28          PULLC    PLP
0639 4C 25 06             JMP  AMPND
063C 28          GOROT    PLP
063D 18                   CLC
063E A6 00       ALIGN    LDX  PINDX
0640 76 50       RPROD    ROR  PROD,X
0642 CA                   DEX
0643 10 FB                BPL  RPROD
0645 28                   PLP          Retrieve Carry from stack
0646 C6 01                DEC  MBIT    Multiplier fully processed?
0648 D0 D0                BNE  NXTBT   No.  Loop for next bit
064A A6 20       RMPLR    LDX  N       Yes.  Rotate multiplier right
064C CA                   DEX
064D 76 30       RBYTE    ROR  MPLR,X
064F CA                   DEX
0650 10 FB                BPL  RBYTE
0652 60                   RTS
```

Example 4. A multiprecision unsigned multiplication subroutine.

✓See List of Advertisers on page 210

=224. In summary, then, *the subroutine in Example 4 can multiply a multiplicand up to 244 bytes long by a multiplier up to 31 bytes long, to yield a product that can be up to 255 bytes long.*

Multiplying Signed Numbers

Subroutines that have been developed in the preceding parts of this article can be used to multiply signed numbers as well as unsigned numbers, provided that the signed multiplier and multiplicand are both positive. In other words, the preceding subroutines can be used to multiply *non-negative* integer numbers. However, many applications require a negative multiplicand to be multiplied by a positive multiplier (or vice versa), or a negative multiplicand to be multiplied by a negative multiplier.

The signs of the multiplier and multiplicand present no problem if you are multiplying decimal numbers with pencil and paper, because you can simply attach a minus sign to the answer if either of the operands was negative! Unfortunately, things don't go that easily if you are multiplying two's complement binary numbers.

Pencil-and-Paper Two's Complement Multiplication

To see the problems you get into

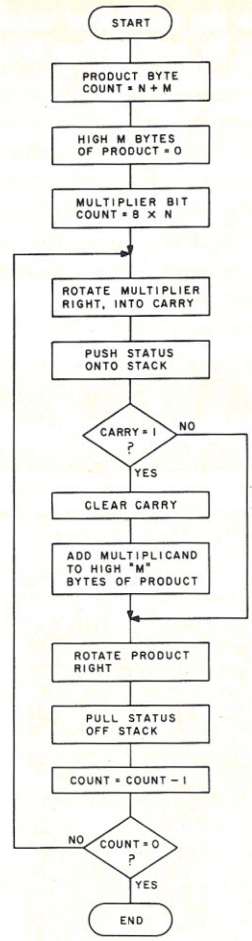


Fig. 4. A multiprecision unsigned multiplication algorithm.

This article deals with multiplication operations on signed and unsigned binary numbers. In an unsigned number, each data bit carries a certain binary weight, according to its position within the number. Within each byte, data bits are numbered from right to left, with the rightmost bit labeled as bit 0 and the leftmost bit labeled as bit 7.

This bit numbering scheme has a direct correlation to the binary weights in that bit 0 has a weight of 2^0 (decimal 1), bit 1 has a weight of 2^1 (decimal 2), and so on. Thus, bit 7 has a weight of 2^7 (decimal 128). The assignments can be summarized as follows:

7	6	5	4	3	2	1	0	Bit position
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Binary weight
128	64	32	16	8	4	2	1	Equivalent decimal weight

As you can see, a single byte can represent an unsigned number from 0 (binary 00000000) to decimal 255 (binary 11111111).

In a signed number, the seven low-order bits (bits 0 through 6) represent data, and have the same relative weights as the bits in unsigned numbers. The most-significant bit (bit 7) represents the sign of the number. If the number is positive or zero, bit 7 is a logic 0. If the number is negative, bit 7 is a logic 1. A single byte can represent a positive signed number from 0 (binary 00000000) to +127 (binary 01111111), or a negative signed number from −1 (binary 11111111) to −128 (binary 10000000).

Why is −1 represented by binary 11111111, rather than by 10000001? The answer is that negative signed numbers are represented in two's complement form. The two's complement form was introduced to eliminate the problems that are associated with allowing zero to be represented in two separate forms, binary combination 00000000 (the positive form) and binary combination 10000000 (the negative form). Using two's complement, zero is represented by only one form, the binary combination 00000000.

To derive the negative two's complement form of a binary number, you simply take the positive form of the number, reverse the sense of each bit (change each 1 to a 0, and each 0 to a 1) and add 1 to the result. The following example shows the steps required in deriving the binary representation of −32 in two's complement form:

+00100000	+32
11011111	One's complement
+ 1	Add 1
11100000	−32 in two's complement form

From the book 6502 Software Design *by Leo J. Scanlon. It is reproduced here with permission of the publisher, Howard W. Sams & Co., Inc.*

✓See List of Advertisers on page 210

multiplying with two's complement negative numbers, let's work out our 103 times 124 example once more, but with a negative multiplier (−103). The pencil-and-paper version will look like this:

```
         01111100  Multiplicand (= +124)
       × 10011001  Multiplier (= −103)
         01111100
        00000000
       00000000
      01111100
     01111100
    00000000
   00000000
  01111100
0100101000011100  Product (= +18972)
```

Not only is this answer too large (recall that the correct magnitude is 12772), but it has the wrong sign as well! Incidentally, the situation does not improve if you multiply −124 by −103. That multiplication will give you a product of +20196.

What, then, can be done to obtain a correct product when we want to multiply negative numbers? Certainly, one valid solution would be to take the two's complement of the negative operand(s), then multiply these two now-positive numbers. If just one of the operands was negative, the resulting product must be two's complemented. If both of the operands were negative, the (positive) product is correct as it stands.

Booth's Algorithm for Signed Multiplication

A much faster solution, and one that does not require either operand to be altered nor the product to be adjusted, is to use a method called Booth's Algorithm. This algorithm is implemented in many of the multiplier chips on the market, and is fully described in Advanced Micro Devices' *Digital Signal Processing Handbook* (John R. Mick, "Understanding Booth's Algorithm in 2's Complement Digital Multiplication," pp. 5–23 through 5–27).

Booth's Algorithm takes advantage of the fact that a string of zeroes in the multiplier requires no additions, but just shifting the partial product, and that a string of ones running from binary weight 2^r to weight 2^s represents a multiplier of $2^{s+1} - 2^r$. For example, if the multiplier = 00001110, then r = 1 and s = 3 and $2^4 - 2^1 = 14$. Note that whereas our previously-described "add-and-shift" algorithm requires three additions for this example, Booth's Algorithm requires only two operations: an addition at weight 2^{s+1} and a subtraction at weight 2^r.

So, when a multiplier is being processed in right-to-left (least-significant bit to most-significant bit) order, Booth's Algorithm boils down to this:

Subtract the multiplicand from the partial product when you find the

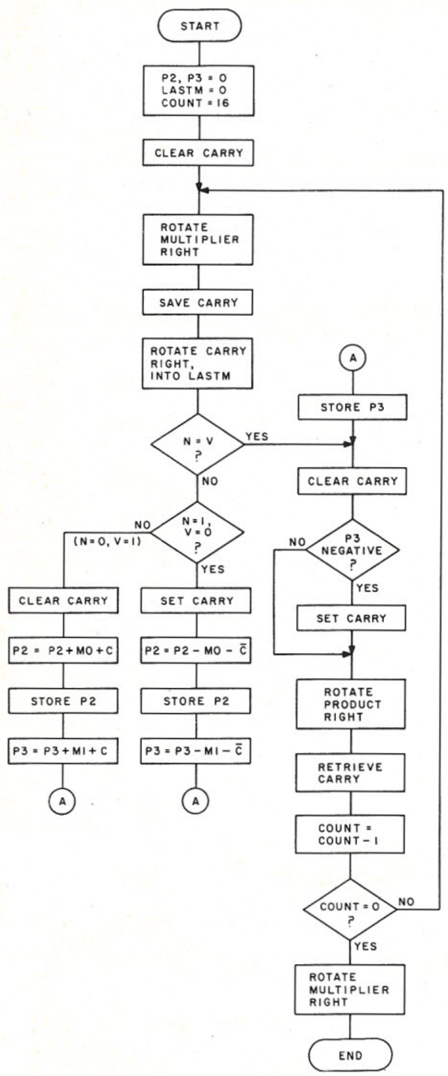


Fig. 5. A double-precision signed multiplication algorithm.

```
This subroutine multiplies a 16-bit signed multiplicand (MPDN,
MPND+1) by a 16-bit signed multiplier (MPLR, MPLR+1), and
returns the 32-bit product in locations PROD (low byte), PROD+1,
PROD+2 and PROD+3 (high byte).

0000                       MPLR=$20      Multiplier
0000                       MPND=$22      Multiplicand
0000                       PROD=$24      Product
0000                       LASTM *=*+1   Previous multiplier bit
0000                       *=$400
0400 A9 00       MLT16S    LDA  #0       Clear P2 and P3 of product
0402 85 26                 STA  PROD+2
0404 85 27                 STA  PROD+3
0406 85 00                 STA  LASTM    Clear LASTM
0408 A2 10                 LDX  #16      Multiplier bit count = 16
0408 18                    CLC
040B 20 46 04    NXTBT     JSR  RMPLR    Go get next multiplier bit
040E 08                    PHP            and save it (as Carry)
040F 66 00                 ROR  LASTM     and put it in LASTM
0411 24 00                 BIT  LASTM    This bit (N) = last bit (V)?
0413 10 10                 BPL  CHKPOS
0415 70 1D                 BVS  ALIGN
0417 38                    SEC           No. N = 1 and V = 0, so
0418 A5 26                 LDA  PROD+2    subtract multiplicand
041A E5 22                 SBC  MPND
041C 85 26                 STA  PROD+2
041E A5 27                 LDA  PROD+3
0420 E5 23                 SBC  MPND+1
0422 4C 32 04              JMP  SP3
0425 50 0D       CHKPOS    BVC  ALIGN
0427 18                    CLC           No. N = 0 and V = 1, so
0428 A5 26                 LDA  PROD+2    add multiplicand
042A 65 22                 ADC  MPND
042C 85 26                 STA  PROD+2
042E A5 27                 LDA  PROD+3
0430 65 23                 ADC  MPND+1
0432 85 27       SP3       STA  PROD+3   Store product MSBY (P3)
0434 18          ALIGN     CLC
0435 24 27                 BIT  PROD+3   P3 negative?
0437 10 01                 BPL  RPROD    No.  Continue with Carry = 0
0439 38                    SEC           Yes.  Set Carry = 1
043A 66 27       RPROD     ROR  PROD+3   Rotate product right
043C 66 26                 ROR  PROD+2
043E 66 25                 ROR  PROD+1
0440 66 24                 ROR  PROD
0442 28                    PLP           Retrieve Carry from Stack
0443 CA                    DEX           Decrement bit count
0444 D0 C5                 BNE  NXTBT    Loop until 16 bits are done
0446 66 21       RMPLR     ROR  MPLR+1   Rotate multiplier right
0448 66 20                 ROR  MPLR
044A 60                    RTS
```

Example 5. A double-precision signed multiplication subroutine.

first 1 in a string of 1's, *add* the multiplicand to the partial product when you find the first 0 in a string of 0's, and do nothing when the bit is identical to the previous multiplier bit.

Mathematically, the basic algorithm as developed by Booth is as follows: y_i is the i-th most significant bit of an n-bit multiplier. y_0 is the least-significant bit and y_{n-1} is the most-significant (sign) bit. So that we can perform a comparison on y_0, we will also define an imaginary less-than-least-significant bit, y_{-1}, and give it a value of 0. X is the multiplicand. Starting with $i = 0$, bit pairs y_i and y_{i-1} are compared:

1) If $y_i = y_{i-1}$, do nothing.

2) If $y_i = 1$ and $y_{i-1} = 0$, subtract X (the multiplicand) from the partial product.

3) If $y_i = 0$ and $y_{i-1} = 1$, add X to the partial product.

As with our unsigned "add-and-shift" multiplications, once the partial product has received the contribution of a particular multiplier bit, it must be right-shifted one bit position. Here, since two's complement numbers are being multiplied, *the partial product's sign must be preserved during the shift*. In other words, the most-significant bit of the partial product must have the same sense after the shift as it did before the shift.

Signed Multiplication Subroutines

With this groundwork, let's see how Booth's Algorithm can be applied to a couple of "real" signed multiplication subroutines.

Fig. 5 is a flowchart for a double-precision (16-bit by 16-bit) signed multiplication subroutine, based on Booth's Algorithm. The initialization box of this flowchart includes a new parameter, LASTM, which is a memory location that will hold the value

Example 6. A multiprecision signed multiplication subroutine.

```
This subroutine multiplies two variable-length, signed integers:
an N-byte multiplier, starting at location MPLR, and an M-byte
multiplicand, starting at location MPND.  The product will be
returned in memory starting at location PROD.  The A, X and Y
registers are affected by the subroutine.

0000                     N=$20        Multiplier length
0000                     M=$21        Multiplicand length
0000                     MPLR=$30     Multiplier location
0000                     MPND=$40     Multiplicand location
0000                     PROD=$50     Product location
0000                     LASTM *=*+1  Previous multiplier bit
0000                     PINDX *=*+1  Product index
0000                     MBIT *=*+1   Multiplier bit count
0000                     *=$600
0600 18         MMPYS    CLC          Calculate product index
0601 A5 20               LDA  N        (N+M-1)
0603 AA                  TAX
0604 65 21               ADC  M
0606 A8                  TAY           and save it in Y
0607 88                  DEY
0608 84 01               STY  PINDX    and in PINDX
060A A9 00               LDA  #0
060C 85 00               STA  LASTM   Clear LASTM and
060E 99 50 00   CLRP     STA  PROD,Y   high M bytes of product
0611 88                  DEY
0612 C4 20               CPY  N
0614 B0 F8               BCS  CLRP
0616 8A                  TXA          Calculate multiplier bit count
0617 0A                  ASL  A
0618 0A                  ASL  A
0619 0A                  ASL  A
061A 85 02               STA  MBIT     and save it in MBIT
061C 18                  CLC
061D 20 79 06   NXTBT    JSR  RMPLR   Get next multiplier bit
0620 08                  PHP           and save it (as Carry)
0621 66 00               ROR  LASTM    and in high bit of LASTM
0623 24 00               BIT  LASTM   This bit (N) = last bit (V)?
0625 10 20               BPL  CHKPOS
0627 70 3E               BVS  ALIGN
0629 A4 20               LDY  N       No.  N = 1 and V = 0, so
062B A2 00               LDX  #0       subtract multiplicand from
062D 38                  SEC           high M bytes of product
062E B9 50 00   SMPND    LDA  PROD,Y
0631 F5 40               SBC  MPND,X
0633 99 50 00            STA  PROD,Y
0636 C8                  INY
0637 E8                  INX
0638 08                  PHP
0639 C4 01               CPY  PINDX
063B F0 06               BEQ  PULLC
```

More

Example 6 continued.

```
063D 90 04              BCC   PULLC
063F 28                 PLP
0640 4C 67 06           JMP   ALIGN
0643 28        PULLC    PLP
0644 4C 2E 06           JMP   SMPND
0647 50 1E     CHKPOS   BVC   ALIGN
0649 A4 20              LDY   N         No.  N = 0 and V = 1, so
064B A2 00              LDX   #0         add multiplicand to
064D 18                 CLC              high M bytes of product
064E B9 50 00  AMPND    LDA   PROD,Y
0651 75 40              ADC   MPND,X
0653 99 50 00           STA   PROD,Y
0656 C8                 INY
0657 E8                 INX
0658 08                 PHP
0659 C4 01              CPY   PINDX
065B F0 06              BEQ   PULLC1
065D 90 04              BCC   PULLC1
065F 28                 PLP
0660 4C 67 06           JMP   ALIGN
0663 28        PULLC1   PLP
0664 4C 4E 06           JMP   AMPND
0667 18        ALIGN    CLC
0668 A6 01              LDX   PINDX
066A B5 50              LDA   PROD,X    Partial product negative?
066C 10 01              BPL   RPROD     No.  Continue with Carry = 0
066E 38                 SEC             Yes.  Set Carry = 1
066F 76 50     RPROD    ROR   PROD,X    Rotate product right
0671 CA                 DEX
0672 10 FB              BPL   RPROD
0674 28                 PLP             Retrieve Carry from stack
0675 C6 02              DEC   MBIT      Multiplier fully processed?
0677 D0 A4              BNE   NXTBT     No.  Loop for next bit
0679 A6 20     RMPLR    LDX   N         Yes.  Rotate multiplier right
067B CA                 DEX
067C 76 30     RBYTE    ROR   MPLR,X
067E CA                 DEX
067F 10 FB              BPL   RBYTE
0681 60                 RTS
```

of the current multiplier bit in bit 7 and the value of the previous multiplier bit in bit 6. These particular bits were selected because they're readily accessible with the 6502's BIT instruction, which puts bits 7 and 6 into the status register's N (Negative) and V (Overflow) flags, respectively.

Example 5 shows a subroutine that follows Fig. 5's flowchart to multiply two 16-bit signed numbers. Similarly, Example 6 shows a subroutine that multiplies two multiprecision signed numbers, an N-byte multiplier and an M-byte multiplicand. Note that Examples 5 and 6 represent the signed multiplication counterparts of the unsigned multiplication subroutines in Examples 3 and 4, respectively. And, incidentally, there's no reason you could not use the subroutines in Examples 5 and 6 to multiply unsigned numbers as well as signed numbers. Booth's Algorithm is applicable to either type of data, and if your multiplier contains long strings of 1's, the longer Booth's Algorithm program will probably perform an unsigned multiplication faster than the "add-and-shift" algorithm program.■

OSI SOFTWARE FOR OSI OSI

A JOURNAL FOR OSI USERS!!

The Aardvark Journal is a bimonthly tutorial for OSI users. It features programs customized for OSI and has run articles like these:

1) Using String Variables.

2) High Speed Basic On An OSI.

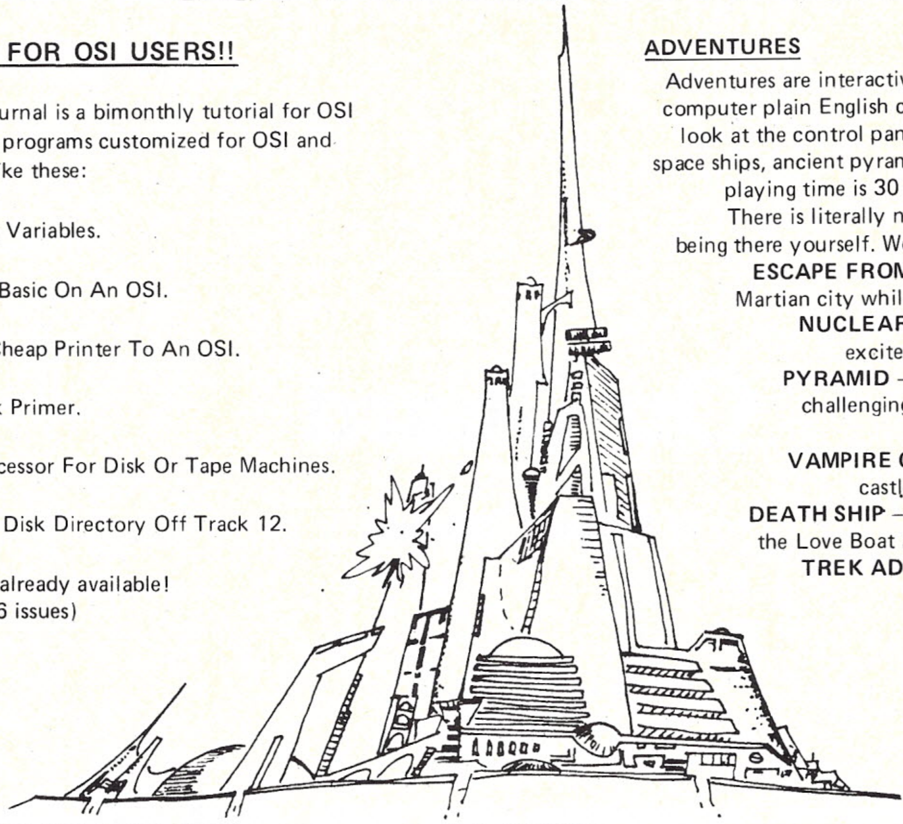
3) Hooking a Cheap Printer To An OSI.

4) An OSI Disk Primer.

5) A Word Processor For Disk Or Tape Machines.

6) Moving The Disk Directory Off Track 12.

First year issues already available!
$9.00 per year (6 issues)



ADVENTURES

Adventures are interactive fantasies where you give the computer plain English commands (i.e. take the sword, look at the control panel.) as you explore alien cities, space ships, ancient pyramids and sunken subs. Average playing time is 30 to 40 hours in several sessions. There is literally nothing else like them – except being there yourself. We have six adventures available.

ESCAPE FROM MARS – Explore an ancient Martian city while you prepare for your escape.

NUCLEAR SUBMARINE – Fast moving excitement at the bottom of the sea.

PYRAMID – Our most advanced and most challenging adventure. Takes place in our own special ancient pyramid.

VAMPIRE CASTLE – A day in old Drac's castle. But it's getting dark outside.

DEATH SHIP – It's a cruise ship – but it ain't the Love Boat and survival is far from certain.

TREK ADVENTURE – Takes place on a familiar starship. Almost as good as being there.

NEW SUPPORT ROMS FOR BASIC IN ROM MACHINES

C1S – for the C1P only, this ROM adds full screen edit functions (insert, delete, change characters in a basic line). Softwave selectable scroll windows, two instant screen clears (scroll window only and full screen), software chose of OSI or standard keyboard format, Bell support, 600 Baud cassette support, and a few other features. It plugs in in place of the OSI ROM. NOTE: this ROM also supports video conversions for 24, 32, 48 or 64 characters per line. Replaces video swap tape on C1P model 2. All that and it sells for a measly $39.95.

C1E/C2E for C1/C2/C4/C8 Basic in ROM machines. This ROM adds full screen editing, software selectable scroll windows, keyboard correction (software selectable), and contains an extended machine code monitor. It has breakpoint utilities, machine code load and save, block memory move and hex dump utilities. A must for the machine code programmer replaces OSI support ROM. Requires installation of additional chip when installed in a C2 or C4. C1 installation requires only a jumper move. Specify system $59.95.

DISK UTILITIES

SUPER COPY – Single Disk Copier
This copy program makes multiple copies, copies track zero, and copies all the tracks that your memory can hold at one time – up to 12 tracks at a pass. It's almost as fast as dual disk copying. – $15.95

MAXIPROSS (WORD PROCESSOR) – 65D polled keyboard only - has global and line edit, right and left margin justification, imbedded margin commands, choice of single, double or triple spacing, file access capabilities and all the features of a major word processor – and it's only $39.95.

P.C. BOARDS

MEMORY BOARDS!! – for the C1P. – and they contain parallel ports!
Aardvarks new memory board supports 8K of 2114's and has provision for a PIA to give a parallel ports! It sells as a bare board for $29.95. When assembled, the board plugs into the expansion connector on the 600 board. Available now!

PROM BURNER FOR THE C1P – Burns single supply 2716's. Bare board – $24.95.

MOTHER BOARD – Expand your expansion connector from one to five connectors or use it to adapt our C1P boards to your C4/8P. - $14.95.

ARCADE AND VIDEO GAMES

GALAXIA one of the fastest and finest arcade games ever written for the OSI, this one features rows of evasive, hardhitting, dogfighting aliens thirsty for your blood. For those who loved (and tired of) Alien Invaders. – P.S. The price is a giveaway. SPECIFY SYSTEM!
Cassette $9.95 – Disk $12.95

TIME TREK (8K) – real time Startrek action. See your torpedoes move across the screen! Real graphics – no more scrolling displays. $9.95

INTERCEPTOR C1P ONLY! An all machine code program as fast and smooth as the arcades. You use your interceptor to protect your cities from hordes of enemy invaders. A pair of automatic cannons help out, but the action speeds up with each wave of incoming ships. The fastest and most exciting C1P game yet.
C1P Cassette $19.95

MINOS – A game with amazing 3D graphics. You see a maze from the top, the screen blanks, and then you are in the maze at ground level, finding your way through on foot. Realistic enough to cause claustrophobia. – $12.95

SINGLE STEPPER / MONITOR

This is probably the finest debugging tool for machine code ever offered for OSI systems. Its' trace function allows you to single step through a machine code program while it continuously displays the A, X, Y and status registers and the program and stack pointers. You can change any of the registers or pointers or any memory location at any time under program control. It takes well under 1k and can be relocated anywhere in free memory. It is a fine tool for all systems – and the best news of all is the extremely low price we put on it. – Tape $19.95 – Disk $24.95

FOR DISK SYSTEMS – (65D, polled keyboard and standard video only.)

SUPERDISK. Contains a basic text editor with functions similar to the above programs and also contains a renumberer, variable table maker, search and new BEXEC* programs. The BEXEC* provides a directory, create, delete, and change utilities on one track and is worth having by itself. – $24.95 on 5" disk - $26.95 on 8".

AARDVARK IS NOW AN OSI DEALER!

Now you can buy from people who can support your machine.

– THIS MONTH'S SPECIALS –

Superboard II	$279
C1P Model II	429
C4P	749
8K 610 board for C1P	269
Epson MX-80 printer with RS232 installed	595

. . . and we'll include a free Text Editor Tape with each machine!

True 32X32 Video Mod Plans for C1P
(4 Chips $3.00 Crystal Required)
$7.95

This is only a partial listing of what we have to offer. We now offer over 100 programs, data sheets, ROMS, and boards for OSI systems. Our $1.00 catalog lists it all and contains free program listings and programming hints to boot.

VISA

master charge

OSI **Aardvark Technical Services • 1690 Bolton • Walled Lake, MI 48088** OSI
(313) 669-3110

✓91

No drilling or soldering with this Apple II circuit mod that prevents unintentional resets.

On Guard

By Robert M. Hirbernik

If you are an Apple II owner who has been constantly plagued with accidental resets, this circuit will bring you much relief. It simply plugs into the Apple's keyboard connector and is easily removed should the Apple ever require service.

Every time you want to reset the computer, you must press and hold down reset and press ESC.

Circuit Description

The circuit continually monitors the keyboard data lines for the ESC code. Three NAND gates invert bits three, six and seven, which are normally low for ESC, before reaching the eight-input NAND gate. Bits one, two, four and five, which are normally high for ESC, pass directly to the eight-input NAND gate.

The $\overline{\text{RES}}$ line, which goes low whenever RESET is pressed, is inverted before reaching the eight-input NAND gate. The STROBE line, which goes high after any key is pressed, is also connected to the eight-input NAND gate, which guarantees that ESC was pressed *after* reset. Thus, when reset is held down and ESC is pressed, all lines reaching the eight-input NAND gate will be high. The output of the NAND gate then goes low, forcing the $\overline{\text{RES}}$ line low and resetting the computer.

You will notice that a two-input AND gate is used with the eight-input NAND gate because a nine-input NAND gate is required. Another two-input AND gate is used to prevent the ESC code from actually entering the Apple while the Reset-ESC sequence is being used. A 0.1 uF capacitor is used to decouple the 5 V line to ground.

I built the reset guard on a small perfboard using sockets for the ICs. I used two 16-pin sockets, one for Apple's keyboard connector and one for a DIP ribbon cable jumper which connects to Apple's main board keyboard socket. I mounted the unit on the right-hand side of the Apple's case with adhesive mounting strips.

Once you start using this inexpensive (under $10) mod, you will wonder how you ever got along without it.■

Address correspondence to Robert M. Hirbernik, 46 Cornell Circle, Pueblo, CO 81005.

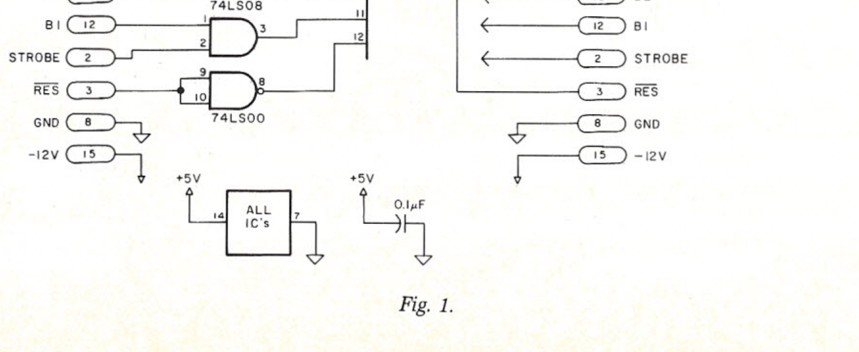


Fig. 1.

The S-50 Bus

By Peter A. Stark

A number of S-50 bus-compatible manufacturers have banded together to spread the word: "Take the S-50 Bus—the Choice is Great!"

It all started at the West Coast Computer Faire in the spring of 1980, when several S-50 bus manufacturers and dealers happened to have adjacent booths. It grew some more at the Philadelphia Computer Show last summer, when 15 S-50 exhibitors gathered together to share a large area of the show set aside only for the S-50 bus. And now there is talk not only of bigger joint shows, but also perhaps even of joint traveling exhibits and joint advertising.

"The time has come," says Richard Don, vice-president of Gimix, Inc., the organizer of the joint gathering at the Philly show and manufacturer of 6800 and 6809 systems on the S-50 bus, "for us to stand up and tell the world we not only exist, but can also offer users just as much as any other bus—if not more.

"People assume that the S-100 bus is better because there are more manufacturers on it. But many of these manufacturers define the bus lines differently resulting in many S-100 buses, while there is only one S-50 bus," he continues. "The S-50 takes second place to no one with the latest state of the art cards. We at Gimix constantly get calls from S-100 users who want to know why we don't make S-100 versions of boards such as our programmable character set video board, or our processor board with an arithmetic processor on it—cards that aren't even available on the S-100 bus. The S-100 may have more software, but S-50 has better development software, and with the 6809 should generate more application packages in the near future."

S-50—The Underdog

Just as the S-100 bus started with one company—Mits/Altair—so did the S-50 bus. It was developed by Southwest Technical Products, Inc. (SWTP), for their 6800-based computer.

Whereas Mits was soon followed by Imsai and others, SWTP had the bus to themselves for a year or two. This had a very important result. When other companies jumped on the S-100 bandwagon, bus standards were very loose, and each company had slightly different ideas about how best to design their equipment. The result was a variety of products which often did not work together. This incompatibility led to the creation of an IEEE standard for future S-100 products.

On the S-50 bus, on the other hand, SWTP was king, and so every company that followed with products for the bus made sure that its products were compatible. As a result, virtually every piece of 6800 hardware or software designed to work on the bus would work with everything else. SWTP ads at the time were pushing the idea that S-50 bus users were bored—they did not have all the problems that users of other systems had.

S-50 hardware and software were both much less expensive than the competition, but this led to a problem. There was a 6800 BASIC which was good as well as cheap—$5 for the 4K BASIC, $10 for the 8K BASIC, almost unheard of prices. Both were excellent BASICs, with ten-digit precision and many other features. And they were reliable. But they were slow.

Since using BASIC was everyone's dream, each processor—and bus, since the S-50 was exclusively a 6800 bus—was judged by the speed of its BASIC. As a result, the 6800 became known as a slow processor.

This was further aggravated by the fact that, because of their design, the 8080 and Z-80 needed faster clocks than a 6800. Even though a 6800 could do as many instructions per second at its 1 MHz clock speed as an 8080 running twice as fast, the numbers game had everyone believing that the 1 MHz 6800 was slower than a 2 MHz 8080.

When Microsoft's 6800 BASIC became available, its users suddenly discovered that this BASIC was even faster than the same BASIC on an 8080, or even the 6502.

But not until Technical Systems Consultants appeared with their 6800 BASIC—which turned out to be the fastest BASIC on any eight-bit processor anywhere—did other users learn that the 6800 was much faster than people had thought. But the damage had been done. The 6800 and, by implication, the S-50 bus had become second best.

Enter the 6809

Suddenly, there is the 6809. Those who know it love it. It's fast, it's powerful enough even for the big-time

Strikes Back!

multiprogramming, multitask users, and it's still simple to use and understand. There are systems with as little as 4K of memory, and there are systems with 384K of RAM, 16 megabytes of disk, and umpteen simultaneous users.

And it's all still on the S-50 bus. Oh yes, there have been some minor changes, but suddenly—voila—the S-50 bus serves both the little guy and the big guns. (When SWTP started, they sold a 2K memory board. Now 64K RAM cards are common, and even a 128K RAM card is available. Just three of those make a 384K system that rivals many minis.)

It doesn't end with the 6809, of course. Though it is the most powerful eight-bit processor yet, plans are already underway to produce 68000 systems on the S-50 bus this year. With the advent of this new Motorola processor, the S-50 bus will have 16-bit computing on a macro scale.

The Variety Is There

You can get a good idea of what's available for the S-50 bus just by glancing at the wares shown at the Philadelphia Computer Show. Here is a quick summary (full names and addresses of S-50 manufacturers are listed at the end of this article).

Thomas Instrumentation was showing off its motherboards, CPU board, I/O boards and especially its new modem card.

Sirius Systems was there to assess interest in its new hard-disk controller.

Control Systems, Inc., was showing its UCSD Pascal system, complete with Pascal compiler and operating system.

Universal Data Research was demonstrating a database management system and application packages for a variety of business applications.

Sonex Systems was showing its Stylograph text editor and processor system, which provides proportional spacing, underlining, half-spacing and more.

Kenyon Microsystems was demonstrating its Forth language and supporting software for implementing this interesting language.

Gimix had the largest booth, which featured complete 6800 and 6809 systems, mainframes, CPU and memory boards, as well as all kinds of I/O equipment. Most interest was shown in their new graphics board and disk controllers.

Microware was demonstrating its new BASIC and an operating system which drew oohs and aahs from those interested in multitasking even on small systems.

Star-Kits (and yours truly) was there, showing a 6802 computer, as well as assorted S-50 bus hardware, software and firmware.

68 Micro Journal sold subscriptions and past issues of their all-68xx magazine.

Hazelwood Computer Systems was demonstrating a new 256×256 graphics card with some quite impressive features.

Frank Hogg Dental Laboratory was there with several new software items, including games, utilities and a job stream control language.

The febe group (with a lowercase "f") showed a low-priced mainframe especially for hobbyists or others for whom a compact, low-cost system is attractive.

Alford and Associates demonstrated a screen editor called SCREDITOR for text editing and processing.

Finally, HHH Enterprises was demonstrating its Spirit language system, as well as various other services.

But Wait . . . There's More

Though the idea of 15 S-50 bus manufacturers getting together and sponsoring a joint exhibit at a major show—complete with talks and user seminars—is unusual enough, it is important to note that even this represented just a small part of the activity on the S-50 bus.

To try to list the products available for the bus would take pages. Here, though, is a quick summary of who makes what; complete names and addresses for all the manufacturers are given at the end of this article.

S-50 mainframes: febe, Gimix, Midwest Scientific, Smoke Signal, Southwest Technical Products.

Motherboards: F&D, febe, Gimix, Midwest Scientific, Percom, Quality, Smoke Signal, Southwest Technical Products, Thomas.

6800 CPU boards: Gimix, Midwest Scientific, Thomas.

6802 CPU boards: F&D, Percom.

6809 CPU boards: F&D, Gimix, Percom, Smoke Signal, Southwest Technical Products.

RAM memory boards: American, Digital Research, Digital Service,

Associate editor Pete Stark (PO Box 209, Mt. Kisco, NY 10549) writes extensively on 68xx systems for Kilobaud Microcomputing.

Gimix, Midwest Scientific, Smoke Signal, Southeastern Microsystems, Southwest Microsystems, Southwest Technical Products, Wimberly, Boaz, Roberts, Hazelwood.

ROM memory boards: Gimix, Micro Works, Southwest Technical Products, Wimberly.

EPROM programmers: Micro Technical Products, Micro Works, Optimal Technology, Southwest Technical Products.

Disk controllers: F&D, Gimix, Midwest Scientific, Peripheral Technology, Smoke Signal, Southwest Technical Products.

Serial I/O: F&D, Gimix, Midwest Scientific, Quality, Southwest Technical Products, Star-Kits.

Parallel I/O: F&D, Gimix, Midwest Scientific, Quality, Southwest Technical Products, Star-Kits, Thomas.

Video boards: F&D, Gimix, Johnson, Percom, Smoke Signal, Thomas.

Graphics boards: Gimix, Hazelwood, Peripheral Technology.

Other I/O boards (such as modems, TV digitizers, A/D converters, prototyping boards and clocks): Avidd, Computer Systems Center, F&D, Gimix, Innovative Technology, JPC, Micro Works, Midwest Scientific, Newtech, Percom, Robertson, Sirius Cybernetics, Thomas, Transition Enterprises, Computerware, Alford, Windrush, Southeastern.

Disk operating systems: Blue Hat, Cer-Comp, Hemenway, Microware, Midwest Scientific, Percom, Smoke Signal, Software Dynamics, Technical Systems Consultants.

Language systems (compilers and interpreters):

- BASIC—Computerware, Microware, Microsoft, Midwest Scientific, Percom, Smoke Signal, Software Dynamics, Technical Systems Consultants.
- COBOL—Smoke Signal, Technical Systems Consultants.
- Forth—Kenyon.
- Fortran—Smoke Signal.
- Lisp—Microware.
- Pascal—Control Systems, Dynasoft, Lucidata, Phillips, Technical Systems Consultants, Omegasoft, Tallgrass.
- Pilot—Southwest Technical Products, Micropi.
- Spirit—(a Forth-type language)—HHH.
- SPL/M—Programma.
- Strubal—Hemenway.

Assemblers: Cer-comp, Cincitek, Great Plains, Hemenway, Microware, Percom, Software Dynamics, Technical Systems Consultants.

Text editors: AAA, Alford, Cer-comp, Micropi, Microware, Midwest Scientific, Percom, Programma, Sonex, Southwest Technical Products, Software Dynamics, Technical Systems Consultants.

Text processors: AAA, Midwest Scientific, Programma, Sonex, Technical Systems Consultants.

Disassemblers: Computer Systems Center, Great Plains.

ROM monitors: F&D, Gimix, Great Plains, Microware, Midwest Scientific, Percom, Sansaska, Sonex, Southwest Technical Products, Star-Kits, Thomas, Technical Systems Consultants.

Other systems software: AAA, Cincitek, Computer Systems Center, Computerware, Hemenway, Frank Hogg, LSI, Microware, Midwest Scientific, Ripley, Southwest Technical Products, Star-Kits, Technical Systems Consultants.

Application programs: AAA, Applevalley, Computerware, DP Systems, Great Plains, Magnusen, Midwest Scientific, Mycroftware, Omnitronics, Schiele, Star-Kits, Technical Systems Consultants, Universal Data, Washington Computer, Westchester.

Games: Alford, Computerware, Frank Hogg, Microware, Star-Kits, Technical Systems Consultants, Mark, Application.

Magazines: 68 Micro Journal, S-50 Newsletter. (And, of course, *Kilobaud Microcomputing*, which regularly features articles on S-50 systems.)

Expandability and Low Price

S-50 systems traditionally have a ROM monitor which replaces a front panel, permitting all the operations which a front panel might otherwise perform (and more), but from a CRT terminal or teleprinter. The need for a terminal seemed like an additional expense at the beginning, especially as compared to the all-in-one appliance computers.

But now a number of manufacturers offer video boards which, along with an inexpensive keyboard, replace the terminal and provide even faster operation. This brings S-50 systems down in price to match even the appliance computers.

The key is expandability. A starter system of the appliance class, such as a TRS-80 or PET, can be cheaper than an equivalent S-50 system. But since S-50 systems are easily—and inexpensively—expanded, a moderate size S-50 system can be much more economical than a moderate system of the appliance class. Moreover, the bus/motherboard concept makes all the accessories fit inside one cabinet, instead of being spread around in expansion adapters and connected with a mass of cable. The result is a more reliable and efficient system as well.

So when you look at system buses, consider the S-50 bus. It has a lot going for it.■

S-50 Bus Manufacturers

AAA Chicago Computer Center, 120 Chestnut Lane, Wheeling, IL 60090

Alford and Associates, PO Box 6743, Richmond, VA 23230

American Computer Works, PO Box 32874, Oklahoma City, OK 73123

Applevalley Day School, 11103 Sagepark Lane, Houston, TX 77089

Application Services Co., PO Box 12227, Wichita, KS 67277.

Avidd Electronics Co., 2210 Bellflower Blvd., Long Beach, CA 90815
Blue Hat Software Co., Box 4127, Flint, MI 48504.
Boaz Co., Box 18081, San Jose, CA 95158.
Cer-Comp, 5566 Ricochet Ave., Las Vegas, NV 89110
Cincitek Software, Box 19365, Cincinnati, OH 45219
Computer Systems Center, 13461 Olive Blvd., Chesterfield, MO 63017
Computer Systems Consultants, Inc., 1415 Latta Lane, Conyers, GA 30207
Computerware Software Services, 1512 Encinitas Blvd., Encinitas, CA 92024
Compuware Corp., PO Box 2710, Cherry Hill, NJ 08003
Control Systems, Inc., 1317 Central, Kansas City, KS 66102
Digital Research Computers, PO Box 401565, Garland, TX 75040
Digital Service and Design, PO Box 741, Newark, OH 43055
DP Systems, PO Box 567, Collegedale, TN 37315
Dynasoft Systems Ltd., PO Box 51, Windsor Jct., Nova Scotia, Canada B0N 2V0
F&D Associates, 1210 Todd Road, New Plymouth, OH 45654
febe group, 51 Hamilton Ave., York PA 17404
Frank Hogg Dental Laboratory, 700 East Water St., Syracuse, NY 13210

Gimix, Inc., 1337 West 37th Place, Chicago, IL 60609
Great Plains Computer Co., Inc., PO Box 916, Idaho Falls, ID 83401
Hazelwood Computer Systems, 7413 N. Lindbergh, Hazelwood, MO 63042
Hemenway Assoc., Inc., 101 Tremont St., Boston, MA 02108
HHH Enterprises, Box 493, Laurel, MD 20810
Innovative Technology, 510 Oxford Park, Garland, TX 75043
Johnson Micro Computer, 2607 E. Charleston, Las Vegas, NV 89104
JPC Products Co., 12021 Paisano Ct., Albuquerque, NM 87112
Kenyon Microsystems, 3350 Walnut Bend, Houston, TX 77042
LSI Enterprises Ltd., PO Box 1227, Woodhaven, NY 11421
Lucidata, Duinsroosweg 10, 2597 KJ The Hague, Netherlands
Gary A. Magnusen, 208 Tinkler, Lafayette, IN 47901
Mark Data Prod., 23802 Barquilla, Mission Viejo, CA 92691
Microsoft, 10800 NE Eighth, Bellevue, WA 98004
Microware Systems Corp., PO Box 4865, Des Moines, IA 50304
Micro Technical Products, 814 W. Keating Ave., Mesa, AZ 85202
Micro Works, PO Box 1110, Del Mar, CA 92014
Micropi, 2445 N. Nugent, Lummi Island, WA 98262
Midwest Scientific Instruments, 220 W. Cedar, Olathe, KS 66061
Mycroftware Systems, PO Box 1138, St. Charles, MO 63301
Newtech Computer Systems, 230 Clinton Street, Brooklyn, NY 11201
Omegasoft, PO Box 70265, Sunnyvale, CA 94086
Omni-Tronics, Inc., Rt. 130, Cranbury, NJ 08512
Optimal Technology, Inc., Blue Wood 127, Earlysville, VA 22936
Percom Data Co., Inc., 211 N. Kirby, Garland, TX 75042
Peripheral Technology Assoc., 290 Lamplighter Lane, Marietta, GA 30067
Phillips Fineware, 876 Plante Dr., Ottawa, Ontario, Canada K1V 9E2
Programma International, 3400 Wilshire Blvd., Los Angeles, CA 90010
Quality Research Co., PO Box 7207, Spokane, WA 99207
Ripley Computers, 126 N. Main St., Souderton, PA 18964
Roberts Control Equipment, 3640 Western Road, Unit 3, Weston, Ontario M9L 1W2, Canada
Robertson Electronics, 1003 Warm Sands Dr., SE, Albuquerque, NM 87123
Sansaska Systems, 3311 Concord Blvd., Concord, CA 94519
Frank Schiele, 1375 Tobias Dr., Chula Vista, CA 92011
Sirius Cybernetics, 7 Euston Place, Leamington Spa, England CV32 4LN
Sirius Systems, 7528 Oak Ridge Hwy., Knoxville, TN 37921
Smoke Signal Broadcasting, 31336 Via Colinas, Westlake Village, CA 91361
Software Dynamics, 2111 W. Crescent, Anaheim, CA 92801
Software Exchange, 2681 Peterboro, W., Bloomfield, MI 48033
Sonex Systems, Box 238, Williamsville, NY 14221
Southeastern Microsystems, PO Box 293, Conyers, GA 30207
Southwest Technical Products Corp., 219 W. Rhapsody, San Antonio, TX 78216
Southwest Microsystems, 6803 Kings Point Lane, Austin, TX 78723
S-50 Newsletter, PO Box 402, Logan, UT 84321
Star-Kits, PO Box 209, Mt. Kisco, NY 10549
Tallgrass Technologies Corp., 7623 W. 86 St., Overland Park, KS 66212
Technical Systems Consultants, Box 2570, West Lafayette, IN 47906
Thomas Instrumentation, 168 Eight St., Avalon, NJ 08202
Transition Enterprises, Inc., Star Route Box 241, Buckeye, AZ 85326
Universal Data Research, Inc., 2457 Wehrle Drive, Buffalo, NY 14221
Walter Wimberly, 2914 Sunrise Drive, Orlando, FL 32803
Washington Computer Services, 3028 Silvern Lane, Bellingham, WA 98225
Westchester Applied Business Systems, PO Box 187, Briarcliff Manor, NY 10510
Windrush Micro Designs, Ltd., Gaymers Way Industrial Estate, North Walsham, Norfolk, England

☆☆☆

4MHZ, DOUBLE DENSITY, COLOR & B/W GRAPHICS . . THE LNW80 COMPUTER

COMPARE THE FEATURES AND PERFORMANCE

FEATURES	LNW80	PMC-80**	TRS-80* MODEL III
PROCESSOR	4.0 MHZ	1,8 MHZ	2.0 MHZ
LEVEL II BASIC INTERP.	YES	YES	LEVEL III BASIC
TRS80 MODEL 1 LEVEL II COMPATIBLE	YES	YES	NO
48K BYTES RAM	YES	YES	YES
CASSETTE BAUD RATE	500/1000	500	500/1500
FLOPPY DISK CONTROLLER	SINGLE/ DOUBLE	SINGLE	SINGLE/ DOUBLE
SERIAL RS232 PORT	YES	YES	YES
PRINTER PORT	YES	YES	YES
REAL TIME CLOCK	YES	YES	YES
24 X 80 CHARACTERS	YES	NO	NO
VIDEO MONITOR	YES	YES	YES
UPPER AND LOWER CASE	YES	OPTIONAL	YES
REVERSE VIDEO	YES	NO	NO
KEYBOARD	63 KEY	53 KEY	53 KEY
NUMERIC KEY PAD	YES	NO	YES
B/W GRAPHICS, 128 X 48	YES	YES	YES
HI-RESOLUTION B/W GRAPHICS, 480 X 192	YES	NO	NO
HI-RESOLUTION COLOR GRAPHICS (NTSC), 128 X 192 IN 8 COLORS	YES	NO	NO
HI-RESOLUTION COLOR GRAPHICS (RGB), 384 X 192 IN 8 COLORS	OPTIONAL	NO	NO
WARRANTY	6 MONTHS	90 DAYS	90 DAYS
TOTAL SYSTEM PRICE	$1,664.00	$1,840.00	$2,187.00
LESS MONITOR AND DISK DRIVE	$1,200.00	$1,375.00	---

When you've compared the features of an LNW80 Computer, you'll quickly understand why the LNW80 is the ultimate TRS80 software compatible system. LNW RESEARCH offers the most complete microcomputer system at an outstanding low price.
We back up our product with an unconventional 6 month warranty and a 10 days full refund policy, less shipping charges.

LNW80 Computer $1,200.00
LNW80 Computer w/B&W Monitor & one 5" Drive $1,664.00
All orders must be prepaid, CA residents please include 6% sales tax.
Shipping and handling charge of $15.00 must be included with every order.
* TRS80 Product of Tandy Corporation.
** PMC Product of Personal Microcomputer, Inc.

LNW80

- BARE PRINTED CIRCUIT BOARD & MANUAL $89.95

The LNW80 - A high-speed color computer totally compatible with the TRS-80*. The LNW80 gives you the edge in satisfying your computation needs in business, scientific and personal computation. With performance of 4 MHz, Z80A CPU, you'll achieve performance of over twice the processing speed of a TRS-80*. This means you'll get the performance that is comparable to the most expensive microcomputer with the compatibility to the world's most popular computer (TRS-80*) resulting in the widest software base.

FEATURES:
. TRS-80 Model 1 Level II Software Compatible
. High Resolution Graphics
  . RGB Output - 384 x 192 in 8 Colors
  . NTSC Video or RF MOD - 128 x 192 in 8 Colors
  . Black and White - 480 x 192
. 4 MHz CPU
. 500/1000 Baud Cassette
. Upper and Lower Case
. 16K Bytes RAM, 12K Bytes ROM
. Solder Masked and Silkscreened

LNW SYSTEM EXPANSION

- BARE PRINTED CIRCUIT BOARD AND MANUAL $69.95
WITH GOLD CONNECTORS $84.95

The System Expansion will allow you to expand your LNW80, TRS-80*, or PMC-80** to a complete computer system that is still totally software compatible with the TRS-80* Model 1 Level II.

FEATURES:
. 32K Bytes Memory
. 5" Floppy Controller
. Serial RS232 20ma I/O
. Parallel Printer
. Real Time Clock
. Screen Printer Bus
. On Board Power Supply
. Solder Masked and Silkscreened

KEYBOARD

LNW80 KEYBOARD KIT $84.95

The Keyboard Kit contains a 63 key plus a 10 key, P.C. board, and remaining components.

LNDoubler + DOS Plus 3.2D

- Assembled and Tested $175.00

Double-density disk storage for the LNW Research's "System Expansion" or the Tandy's "Expansion Interface". The LNDoubler™ is totally software compatible with any double density software generated for the Percom's Doubler***. The LNDoubler™ provides the following outstanding features.

. Store up to 350K bytes on a single 5" disk
. Single and double density data separation
. Precision write precompensation circuit
. Software switch between single and double density
. Hardware override into single density only
. Easy plug in installation requiring no etch cuts, jumpers or soldering
. 35, 40, 77, 80 track 5" disk operation
. 120 day parts and labor Warranty

*** Doubler is a product of Percom Data Company, Inc.

DOS PLUS 3.2D

Micro Systems software's double density disk operating system. This operating system contains all the outstanding features of a well developed DOS, with ease in useability.

LNW DATA SEPERATOR

- Assembled and Tested $17.95

The LNW Data Separator provides you with a reliable and inexpensive means of solving your disk data read error problems for your 5" single density drives. Compatible with both the LNW System Expansion and Tandy's Expansion Interface. Some soldering is required.

CASE

LNW80 CASE . Add $12.00 for postage $84.95

The streamline design of this metal case will house the LNW80, LWN System Expansion, LNW80 Keyboard, power supply and fan, LNDoubler™, or LNW Data Separator. This kit includes all the hardware to mount all of the above.

PARTS AVAILABLE FROM LNW RESERACH
4116 - 200ns RAM
6 chip set $26.00
8 chip set $33.50
16 chip set $64.00
24 chip set $94.00
32 chip set $124.00

. LNW80 "Start up parts set" LNW80-1 . . . . $82.00
. LNW80 "Video parts set" LNW80-2 . . . . $31.00
. LNW80 Transformer LNW80-3 . . . . $18.00
. LNW80 Keyboard cable LNW80-4 . . . . $16.00
. 40 Pin computer to expansion cable $15.00
. System Expansion Transformer $19.00
. Floppy Controller (FD1771) and UART (TR1602) . . . $30.00

LNW RESEARCH ~198
C O R P O R A T I O N
14661-C MYFORD RD.
TUSTIN CA. 92680

VISA & MASTER CHARGE ACCEPTED, Add $3 for postage
ORDERS & INFO. NO. 714-552-8946
SERVICE NO. 714-641-8850

If you enjoyed Messent's cassette save program in our April issue, you'll enjoy this approach to saving OSI machine-language programs on tape.

Dump It on Cassette

By Colin Macauley

After fooling about with your new OSI Challenger 1P or Superboard II in BASIC, you may wish to consider an alternative. Have you ever answered "M" to the "D/C/W/M?" prompt when the machine is turned on and the break key hit? Not many have, judging from the lack of machine-language articles found in the magazines.

If you don't know the difference between an accumulator and a register, some preliminary reading will be required (see first reference). Otherwise, shame on you for not fully using your machine. To appreciate the instruction set of the 6502, its advantages and limitations, I suggest perusal of the classic textbooks (see references).

The three pages devoted to machine language in the OSI User's Manual may be devastating and may be the reason you have never tried it before. The following commands are at your disposal:

Address Mode Commands

/—Change to Data Mode.

G—Commence execution of program at address shown on screen.

L—Load memory from audio cassette.

Data Mode Commands

●—Change to Address Mode.

"RETURN"—Increment memory address shown on screen.

From these commands you will note that the OSI machine-language monitor is very basic and has a striking anomaly. You can load from cassette, but there is no command to let you dump your program to cassette.

The program listing shows how to save your machine-language programs on tape. The program is located at $1B60, but may be located at any convenient address if the lines with asterisks are altered to suit your address changes. The following zero-page addresses are used:

$F0—START—low byte of starting address of program to be saved.

$F1—START+1—high byte of above starting address.

$F2—END—low byte of the end address of program to be saved.

$F3—END+1—high byte of above end address.

$F4—TEMP— working variables.

$F5—TEMP+1— working variables.

$F6—COUNT—counter variable.

The data for memory locations $F0 through $F3 must be entered before saving a program.

The operation of the program is very simple. The data is converted to ASCII and then dumped to cassette via the Motorola 6850 ACIA. Additional comments for the program lines are provided in Table 1.

To demonstrate the program, type in the hexadecimal listing commencing at location $1B60. Enter starting address of program in zero-page; i.e., $00F0 contains $60 and $00F1 contains $1B (always remember that the low address byte is stored first in 6502 assembly language). Similarly, the end address is entered; i.e., $00F2 contains $32 and $00F3 contains $1C. Turn your recorder on and run the program by entering ".1B60G". The dump is completed when the screen indicates a return to the monitor by displaying $0000.■

01–04	—Initialize ACIA ($F000, $F001).
05–12	—Dump address "header" to cassette.
13–19	—Check whether more than one page is to be dumped. Go to LESS if less than one page involved.
20–43	—Dump whole pages to cassette and jump to REST when complete.
44–47	—Less than one page to be dumped.
48–66	—Dump remainder of incomplete page or, if less than one page, dump whole program.
67–79	—Dump address "footer" (which enables return to monitor) to cassette.
80–97	—Converts a byte to ASCII and dumps ASCII to cassette.
98–105	—Checks that ACIA is ready to transmit and dumps ASCII to cassette.

Table 1. Program line comments.

References

Osborne, Adam. *An Introduction to Microcomputers*. Osborne/McGraw-Hill. Volumes 0 and 1.

Zaks, Rodney. *Programming the 6502*. Sybex.

Leventhal, L. A. *6502 Assembly Language Programming*. Osborne/McGraw-Hill.

Address correspondence to Colin Macauley, c/o Callinan and Associates, PO Box 238, East Melbourne, Victoria 3002, Australia.

Machine-language dump program for the Superboard II or Challenger C1P.

```
($1B60)             LDA #$03        A9,03
                    STA $F000       8D,00,F0
                    LDA #$B1        A9,B1
                    STA $F000       8D,00,F0
                    LDA "."         A9,2E
                    JSR PUNCH       20,26,1C
                    LDA START+1     A5,F1
                    JSR ASCII       20,0B,1C
                    LDA START       A5,F0
                    JSR ASCII       20,0B,1C
                    LDA "/"         A9,2F
                    JSR PUNCH       20,26,1C
                    CLD             D8
                    SEC             38
                    LDA END         A5,F2
                    SBC START       E5,F0
                    LDA END+1       A5,F3
                    SBC START+1     E5,F1
                    BEQ LESS        F0,32
                    TAX             AA
                    LDA START       A5,F0
                    STA TEMP        85,F4
                    LDA START+1     A5,F1
                    STA TEMP+1      85,F5
                    LDY #$ØØ        A0,00
REDO ($1B95)        LDA #$ØØ        A9,00
                    STA COUNT       85,F6
LOOP ($1B99)        LDA (TEMP),Y    B1,F4
                    JSR ASCII       20,0B,1C
                    LDA "CR"        A9,0D
                    JSR PUNCH       20,26,1C
                    INC TEMP        E6,F4
                    BNE COUNT       D0,02
                    INC TEMP+1      E6,F5
COUNT               INC COUNT       E6,F6
                    LDA #$ØØ        A9,00
                    CMP COUNT       C5,F6
                    BEQ NEXTPAGE    F0,03
                    JMP LOOP        4C,99,1B
NEXTPAGE
($1BB4)             DEX             CA
                    CPX #$ØØ        E0,00
                    BEQ REST        F0,0B
                    JMP REDO        4C,95,1B
LESS ($1BBC)        LDA START       A5,F0
                    STA TEMP        85,F4
                    LDA START+1     A5,F1
                    STA TEMP+1      85,F5
REST ($1BC4)        LDA END         A5,F2
                    SEC             38
                    SBC START       E5,F0
                    TAX             AA
                    INX             E8
                    LDY #$ØØ        A0,00
                    LDA #$ØØ        A9,00
                    STA COUNT       85,F6
LOOPA ($1BD1)       LDA (TEMP),Y    B1,F4
                    JSR ASCII       20,0B,1C
                    LDA "CR"        A9,0D
                    JSR PUNCH       20,26,1C
                    INC TEMP        E6,F4
                    BNE COUNTA      D0,02
                    INC TEMP+1      E6,F5
COUNTA              INC COUNT       E6,F6
                    CPX COUNT       E4,F6
                    BEQ OUT         F0,03
                    JMP LOOPA       4C,D1,1B
OUT                 LDA "."         A9,2E
                    JSR PUNCH       20,26,1C
                    LDA "F"         A9,46
                    JSR PUNCH       20,26,1C
                    LDA "E"         A9,45
                    JSR PUNCH       20,26,1C
                    LDA "Ø"         A9,30
                    JSR PUNCH       20,26,1C
                    LDA "Ø"         A9,30
                    JSR PUNCH       20,26,1C
                    LDA "G"         A9,47
                    JSR PUNCH       20,26,1C
                    JMP $FEØØ       4C,00,FE
ASCII ($1C0B)       PHA             48
                    LSR A           4A
                    LSR A           4A
                    LSR A           4A
                    LSR A           4A
                    SED             F8
                    CLC             18
                    ADC #$9Ø        69,90
                    ADC #$4Ø        69,40
                    JSR PUNCH       20,26,1C
                    PLA             68
                    AND #$ØF        29,0F
                    CLC             18
                    ADC #$9Ø        69,90
                    ADC #$4Ø        69,40
                    JSR PUNCH       20,26,1C
                    CLD             D8
                    RTS             60
PUNCH ($1C26)       PHA             48
LOOPB               LDA $FØØØ       AD,00,F0
                    LSR A           4A
                    LSR A           4A
                    BCC LOOPB       90,F9
                    PLA             68
                    STA $FØØ1       8D,01,F0
($1C32)             RTS             60
```

Here's an intriguing kid's game program written for the SWTP, but in fairly standard BASIC, with no specialities of the system.

Number Squares

By Marc I. Leavey

It all started when my five-year-old daughter returned from a birthday party with a small plastic square with a four-by-four grid, in which there are 15 numbered tiles and one blank square. The object of this number squares game is to move the tiles one at a time until they are in the desired order. It may not seem like much, but it's fun, and it even had me going for a while.

Did I say it "had me going"? That was the problem—I was becoming addicted. What makes the game so intriguing is that you can only move one tile at a time. It wasn't long, though, before I duplicated the game on my CRT.

In the computer version, the board is represented by a four-by-four array. Shifting a tile into the solitary blank space is the only legal move. The number of the piece to be moved is thus the player input.

The computer checks to see if the piece input is next to a space and, if it is, moves the piece. Moves continue until the pieces are in the desired order, at which time a win is declared.

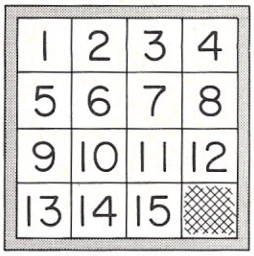


Fig. 1. One sequential solution to Number Squares.

To set up the program, a four by four array is established and sequentially filled with the integers one through 16. The blank is represented by 16, and resides at the lower right.

Next, the board is scrambled by randomly moving the blank 200 times. Each move is up, down, left or right, and, after 200 random moves, the board is quite a mess.

It is important to realize the board is not randomly filled with numbers. If it were, there might be no solution to the puzzle. Instead, the board is scrambled just as a real one is, by starting with a correct setup, and then moving pieces randomly—by the rules—to clear the board.

Once the board is initialized, it is displayed on the screen, and a user input is requested.

The input number is checked for range and validity of movement. If all is in order, the input piece is moved into the blank, and the blank into the position of the input piece. Remember, only vertical or horizontal moves are allowed; no diagonals are permitted.

After each move, the order of the board is scanned and, if in proper order, a win is declared.

But what is the proper order? As originally conceived, a player wins when the tiles are in numerical order (Fig. 1). However, another solution is possible. In the Magic Square, every column and row adds up to 30. In the interest of a more challenging game, I will not provide a diagram of the Magic Square solution. It is available in the program if you look for it, but not knowing it exactly makes the game that much more fun.

I set up the display of the game for a terminal running at 1200 baud, with homing and erasing capabilities. With some creative programming and a VDM, quite a nice display can be constructed and inserted in place of the display module in the program. No fancy features of any special BASIC were used, and the program

Number Squares program listing.

```
0010 : NUMBER SQUARE GAME
0020 : VER 4.0 - 12 NOV 79
0030 : MARC I. LEAVEY, M.D.
0040 LINE= 0
0050 DIGITS= 0
0060 PRINT "N U M B E R   S Q U A R E S"
0070 PRINT "-------------------------------"
0080 PRINT
0090 PRINT "WELCOME TO THE WORLD OF"
0100 PRINT "CONFUSION.  THERE ARE TWO"
0110 PRINT "VERSIONS OF NUMBER SQUARES:"
0120 PRINT " 1 - SEQUENTIAL"
0130 PRINT " 2 - MAGIC SQUARE"
0140 INPUT "WHICH IS YOUR PLEASURE",T
```

More

Marc I. Leavey, M.D., 4006 Winlee Road, Randallstown, MD 21133.

Program listing continued.

```
0150 IF T=1 GOTO 310
0160 IF T<>2 GOTO 140
0170 :
0180 : SET UP MAGIC
0190 : SQUARE BOARD
0200 :
0210 FOR I=1 TO 4
0220 FOR J=1 TO 4
0230 READ M(I,J)
0240 LET B(I,J)=M(I,J)
0250 NEXT J
0260 NEXT I
0270 DATA 1,6,15,8,12,11,2,5,10,13,4,3,7,16,9,14
0280 LET I1=4
0290 LET J1=2
0300 GOTO 440
0310 :
0320 : SET UP SEQUENTIAL
0330 : BOARD
0340 :
0350 DIM B(4,4)
0360 FOR I=1 TO 4
0370 FOR J=1 TO 4
0380 LET B(I,J)=(I-1)*4+J
0390 NEXT J
0400 NEXT I
0410 LET I1=4
0420 LET J1=4
0430 :
0440 : NOW SCRAMBLE THE BOARD
0450 : TWO HUNDRED TIMES
0460 :
0470 PRINT "I AM NOW SCRAMBLING THE BOARD..."
0480 FOR Q=1 TO 200
0490 LET M=INT(1+RND*4)
0500 ON M GOTO 510,560,610,660
0510 IF I1=1 GOTO 490
0520 LET B(I1,J1)=B(I1-1,J1)
0530 LET B(I1-1,J1)=16
0540 LET I1=I1-1
0550 GOTO 700
0560 IF I1=4 GOTO 490
0570 LET B(I1,J1)=B(I1+1,J1)
0580 LET B(I1+1,J1)=16
0590 LET I1=I1+1
0600 GOTO 700
0610 IF J1=1 GOTO 490
0620 LET B(I1,J1)=B(I1,J1-1)
0630 LET B(I1,J1-1)=16
0640 LET J1=J1-1
0650 GOTO 700
0660 IF J1=4 GOTO 490
0670 LET B(I1,J1)=B(I1,J1+1)
0680 LET B(I1,J1+1)=16
0690 LET J1=J1+1
0700 NEXT Q
0710 :
0720 : PRINT BOARD
0730 :
0740 LET M9=0
0750 : OUTPUT A "HOME UP"
0760 PRINT CHR$(16);
0770 PRINT "------------------"
0780 FOR I=1 TO 4
0790 FOR J=1 TO 4
0800 PRINT ": ";
0810 IF B(I,J)=16 PRINT "   ";:GOTO 840
```

More

should run on any machine.

This is not a complicated program, but it shows methods of array handling and move checking. Display formatting and concern for screen readability are also factors.

The game itself is addicting. If you want to make it easier at first, change the setup scramble in line 480 to loop to 50 instead of 200, or make the loop variable and allow the user to choose his own degree of difficulty. A sample routine might be:

```
0480 INPUT "DEGREE OF DIFFICULTY
     (10-500);" Q9
0485 FOR Q=1 TO Q9
```

Try it. And if you've got a little girl around five, don't blame me if she takes over. Mine did! ■

Program listing continued.

```
0820 IF B(I,J)<10 PRINT " ";
0830 PRINT B(I,J);
0840 NEXT J
0850 PRINT ":"
0860 PRINT "-------------------"
0870 NEXT I
0880 :
0890 : ERASE REST OF SCREEN AND
0900 : BEEP FOR INPUT
0910 :
0920 PRINT CHR$(22);CHR$(7);CHR$(7);
0930 :
0940 : INPUT MOVE
0950 :
0960 INPUT "MOVE WHICH PIECE",M
0970 LET I1=0:J1=0
0980 FOR I=1 TO 4
0990 FOR J=1 TO 4
1000 IF B(I,J)=M THEN I1=I:J1=J
1010 NEXT J
1020 NEXT I
1030 IF I1=0  THEN PRINT "I CAN'T FIND THAT NUMBER":GOTO 940
1040 LET I2=0:J2=0
1050 FOR I=I1-1 TO I1+1
1060 IF I>4 GOTO 1090
1070 IF I<1 GOTO 1090
1080 IF B(I,J1)=16 THEN I2=I:J2=J1:GOTO 1170
1090 NEXT I
1100 FOR J=J1-1 TO J1+1
1110 IF J>4 GOTO 1140
1120 IF J<1 GOTO 1140
1130 IF B(I1,J)=16 THEN I2=I1:J2=J:GOTO 1170
1140 NEXT J
1150 LET M9=M9+1
1160 PRINT "NOT A VALID MOVE":GOTO 940
1170 LET B(I2,J2)=M
1180 LET B(I1,J1)=16
1190 ON T GOTO 1210,1320
1200 :
1210 : SEQUENTIAL SOLUTION
1220 :
1230 LET C=0
1240 FOR I=1 TO 4
1250 FOR J=1 TO 4
1260 IF B(I,J)<C GOTO 720
1270 LET C=B(I,J)
1280 NEXT J
1290 NEXT I
1300 PRINT "YOU GOT IT!"
1310 GOTO 1450
1320 :
1330 : MAGIC SQUARE SOLUTION
1340 : CHECK
1350 :
1360 FOR I=1 TO 4
1370 FOR J=1 TO 4
1380 IF B(I,J)<>M(I,J) GOTO 720
1390 NEXT J
1400 NEXT I
1410 :
1420 : A WIN IS DECLARED!
1430 :
1440 PRINT "THAT IS THE CORRECT SOLUTION!"
1450 INPUT "LIKE TO PLAY ANOTHER GAME",I$
1460 IF LEFT$(I$,1)="Y" THEN RUN
1470 END
```

✓See List of Advertisers on page 210

This installment tells you how to add a useful editing function to your computer.

Thoughts on The 68XX Systems

By Garry O. Caudell and Ron Silver

As part of writing this continuing series on 68xx systems, I often receive suggestions on useful topics from avid fans of the 6800 and 6809. This month I'll sit back and combine two contributions from readers into an article under their name.—Pete Stark.

BASIC interpreters usually require that each line be numbered, and automatically insert new lines into line number order, thus providing a certain amount of editing capability.

It is possible to insert, delete or substitute new lines into a program by making use of the line numbers. When a program statement is no longer needed, it can be deleted simply by entering its line number. A line can be changed by retyping a line with the same line number. And finally, a line can be inserted between two existing lines simply by using a line number that is between the values of the line numbers of those two statements.

The above features of BASIC are useful. However, in the process of writing and debugging a BASIC program, you'll often need to make minor modifications to quite a few lines. In most 6800 and 6809 BASICs, each of these modifications requires that you retype the entire line, even when only one or two characters need to be changed.

On disk systems, users often resort to the text editor to make such changes. Yet it is possible to overcome this limitation of the BASIC interpreter by adding to it a more powerful edit function. This modification would make the BASIC system easier to use by letting the user edit a line without retyping the whole line.

Many other computers (such as the TRS-80) have such editing capability in their BASIC, since not much additional code is required.

This article presents the modifications to implement this editing function in SWTP 8K BASIC versions 2.0 and 2.2 (see Listing 1), SWTP Disk BASIC version 3.5 (Listing 2) and Percom Super BASIC (Listing 3).

Although there are differences in the three implementations, the idea is the same. In fact, it is also easily applied to the TSC Text Editor itself (see Listing 4, and further details later). Although the TSC Text Editor provides many ways of editing entire lines, its provisions for manipulating characters within a line are somewhat awkward. This modification fixes that problem.

What It Does

To implement this feature, you must use a CRT terminal. The modification dynamically shows each edit-

Control Character	ASCII Code	Function
Control-E	$05	Enters edit mode—may be used during any input.
Control-R	$12	Recalls a line from memory for editing. The line recalled will be the line whose line number appears at the beginning of the line being typed when control-R is entered.
Control-A	$01	Adds a space at cursor position.
Control-D	$04	Delete character at cursor position.
Control-H	$08	Backspace—move cursor left.
Control-I	$09	Tab—move cursor right.
(CR)	$0D	Carriage return—enters the line into memory as if it had just been just typed in, regardless of cursor position.

Table 1. Summary of edit commands.

Address correspondence to Ron Silver, Kelvin High School, 155 Kingsway Ave., Winnipeg, Man-

ing change as it occurs by rapidly spacing forward or backward to erase and rewrite the changed line. Since most teleprinters do not backspace, this would not work on such a terminal; even with backspacing, it would repeatedly overprint a line and produce a black mess.

The modified BASIC interpreter lets the user insert, delete or change any character in any statement of a program. The edit functions are controlled by six control keys.

BASIC program entry and editing is unaffected until a control-E ($05 code) is typed. This causes a branch to the edit subroutine in the patch. This subroutine then processes all succeeding characters and commands entered from the keyboard until hitting the carriage return returns back to normal BASIC entry.

Editing occurs on the line currently in BASIC's input buffer. If a line is being entered, then that line will be edited. This mode, however, is probably not that useful, since in many cases it is easier to simply backspace to the wrong character and retype from there.

A far more powerful feature of this editor modification is its ability to recall lines previously entered. Typing the number of this line followed by control-E to enter the editor, and then control-R to recall the line, brings a line from memory back into the buffer for editing. The line now appears on the CRT as if it had just been typed, and the cursor is positioned at the end of the line.

At this point, the cursor would normally be moved to the character where a change is to be made.

Backspace (control-H) moves the cursor back to the left without erasing the characters it is stepping over.

Horizontal tab (control-I) moves the cursor to the right, again without erasing the characters it is stepping over.

Once the cursor is positioned where a change is wanted, there are three ways to proceed:

Control-A inserts (adds) a space at the current cursor position and mŏves the following characters one place to the right. On a slow terminal you can see the rest of the line being rewritten; on a fast terminal (9600 baud, for example) this occurs so fast that it is nearly invisible.

itoba, Canada R3M 0G3; or Garry O. Caudell, 3125 Robin Lynn Drive, Ashland, KY 41101.

Listing 1. Editing modification in SWTP 8K BASIC versions 2.0 and 2.2.

```
                          NAM    EDITTOR
                          OPT    0
                   * BASIC ENTRY POINTS
      0485   PSHX   EQU    $0485     PUSH INDEX
      00B0   BUFFER EQU    $B0       BEGINNING OF THE BASIC BUFFER
      00F8   BUFEND EQU    $F8       END OF BUFFER
      0440   BERROR EQU    $0440     ERROR ROUTINE
      0395   INPUTC EQU    $0395     INPUT ROUTINE
      0386   OUTC   EQU    $0386     OUTPUT ROUTINE
      049A   PULX   EQU    $049A     PULL INDEX
      004A   COUNTR EQU    $4A
      0024   BUFPNT EQU    $24       POINTER TO BUFFER POSITION
      002C   MEMPNT EQU    $2C       POINTER TO MEMORY POSITION
      00AC   TEMP   EQU    $AC
      0AC7   BERR2  EQU    $0AC7     ERROR ROUTINE
      0ACA   NUMC   EQU    $0ACA     NUMERIC CONVERSION
      0A87   LINEC  EQU    $0A87     FIND LINE IN MEMORY
      0467   PSTR   EQU    $0467     PRINT STRING ROUTINE
      0453   PLINEC EQU    $0453     BRANCH POINT BACK TO BASIC
                   * MISC CONSTANTS
      0008   BACK   EQU    $08       BACK SPACE CHARACTER
      0009   TAB    EQU    $09       FORWARD TAB CHARACTER
1EAF                ORG    $1EAF
1EAF BD 0485 EDIT   JSR    PSHX
1EB2 6F 00   CLEAR  CLR    ,X        CLEAR BUFFER TO END
1EB4 08             INX
1EB5 8C 00F8        CPX    #BUFEND   IS IT THE END OF THE BUFFER?
1EB8 26 F8          BNE    CLEAR
1EBA BD 049A        JSR    PULX
1EBD BD 0395 INPUT  JSR    INPUTC    INPUT LOOP
1EC0 81 0D          CMP A  #$0D      CHECK FOR CARRIAGE CONTROL
1EC2 26 03          BNE    CONT1
1EC4 7E 1FBD        JMP    ENDRTN    IF SO JUMP BACK TO BASIC
1EC7 81 08   CONT1  CMP A  #BACK     CHECK FOR BACKSPACE
1EC9 27 2D          BEQ    BACKSPC
1ECB 81 09          CMP A  #TAB
1ECD 27 15          BEQ    FORWARD
1ECF 81 04          CMP A  #$04      CONTROL D
1ED1 27 34          BEQ    DELETE
1ED3 81 01          CMP A  #$01      CONTROL A
1ED5 27 5D          BEQ    ADD
1ED7 81 12          CMP A  #$12      CONTROL R
1ED9 26 03          BNE    CONT3
1EDB 7E 1F52        JMP    RECALL
1EDE 81 1F   CONT3  CMP A  #$1F      CHECK IF CONTROL CHARACTER
1EE0 2B DB          BMI    INPUT     IF SO IGNORE
1EE2 A7 00   STORCH STA A  ,X        STORE CHARACTER IN BUFFER
                   * MOVE CURSOR FORWARD
1EE4 A6 00   FORWAR LDA A  ,X
                   *  CHECK IF AT END OF LINE IN BUFFER
1EE6 26 07          BNE    CONT2     BRANCH IF NOT
1EE8 86 08          LDA A  #BACK
                   *  PUT CURSOR BACK WHERE IT BELONGS
1EEA BD 0386        JSR    OUTC

1EED 20 CE          BRA    INPUT
                   *   MOVE TO NEXT POSITION IN BUFFER
1EEF 08      CONT2  INX              MOVE TO NEXT POSITION IN BUFF
1EF0 8C 00F8        CPX    #BUFEND   CHECK FOR END OF BUFFER
1EF3 26 C8          BNE    INPUT
1EF5 7E 0440 ERROR  JMP    BERROR
                   * BACKUP CURSOR
1EF8 8C 00B0 BACKSP CPX    #BUFFER   CHECK IF AT START OF BUFFER
1EFB 27 03          BEQ    ATBEG     BRA IF SO
1EFD 09             DEX
1EFE 20 BD   LOOPBA BRA    INPUT
1F00 86 09   ATBEG  LDA A  #TAB      MOVE CURSOR FORWARD
1F02 BD 0386        JSR    OUTC
1F05 20 B6          BRA    INPUT



                   * DELETE A CHARACTER FROM BUFFER AND CRT
1F07 BD 0485 DELETE JSR    PSHX
                   * ZERO COUNTER IN ORDER TO REPOSITION CURSOR
1F0A 7F 004A        CLR    COUNTR
                   * MOVE CURSOR AND BUFFER CONTENTS ONE POSITION LEFT
1F0D A6 01   MOVELF LDA A  1,X
1F0F BD 0386        JSR    OUTC
1F12 A7 00          STA A  X
1F14 08             INX
1F15 7C 004A        INC    COUNTR    COUNT CHARACTERS TO END
1F18 4D             TST A
1F19 26 F2          BNE    MOVELFT
1F1B 86 20   SPACE  LDA A  #$20      PRINT SPACE
1F1D BD 0386        JSR    OUTC
                   * PUT CURSOR BACK TO CORRECT POSITION
1F20 7D 004A MOVECS TST    COUNTR    CHECK TO SEE IF FINISHED
1F23 27 0A          BEQ    ENDD
1F25 86 08          LDA A  #BACK
1F27 BD 0386        JSR    OUTC
1F2A 7A 004A        DEC    COUNTR
1F2D 20 F1          BRA    MOVECSR
1F2F BD 049A ENDD   JSR    PULX
1F32 20 CA          BRA    LOOPBACK
```

More →

Listing 1 continued.

```
00740 1F34 BD 0485 ADD    JSR    PSHX
00746                     * ADD SPACE TO BUFFER AT CURSOR POSITION
00750 1F37 7F 004A        CLR    COUNTR    ZERO COUNTER
00760 1F3A C6 20          LDA B  #$20      PUT SPACE IN BUFFER
00765                     * MOVE CURSOR AND BUFFER RIGHT ONE PLACE
00770 1F3C 17      MOVERT TBA
00780 1F3D E6 00          LDA B  X
00790 1F3F A7 00          STA A  X
00800 1F41 BD 0386        JSR    OUTC
00810 1F44 08             INX
00820 1F45 8C 00F8        CPX    #BUFEND   CHECK FOR FILLED BUFFER
00830 1F48 27 AB          BEQ    ERROR
00840 1F4A 7C 004A        INC    COUNTR    COUNT CHARACTERS FILLED
00850 1F4D 4D             TST A
00860 1F4E 26 EC          BNE    MOVERT
00870 1F50 20 C9          BRA    SPACE


00877                     * RECALL LINE FROM MEMORY AND PLACE IN BUFFER
00880 1F52 8D 5B   RECALL BSR    ZERCSR    MOVE CURSOR TO START OF LINE
00890 1F54 CE 00B0        LDX    #BUFFER   FIND LINE NUMBER IN BUFFER
00900 1F57 BD 0ACA        JSR    NUMC      CONVERT TO BCD FOR LINEC
00910 1F5A 24 03          BCC    CONT4     INVALID LINE NUMBER?
00920 1F5C 7E 0AC7 ERR2   JMP    BERR2
00925                     *  STORE POSITION OF END OF LINE NUMBER
00930 1F5F DF 24   CONT4  STX    BUFPNT
00940 1F61 BD 0A87        JSR    LINEC     FIND LINE IN MEMORY
00950 1F64 25 F6          BCS    ERR2      BRANCH IF NOT FOUND
00955                     *  SKIP PAST LINE NUMBER STORED IN MEMORY
00960 1F66 08             INX
00970 1F67 08             INX
00975                     *  STORE POSITION OF LINE IN MEMORY
00980 1F68 DF 2C          STX    MEMPNT
00990 1F6A 8D 1F          BSR    STORKY    STORE KEYWORD IN BUFFER
01000 1F6C 08             INX
01005                     * TRANSFER CHARACTERS FROM MEMORY TO BUFFER
01010 1F6D A6 00   SLOOP  LDA A  X
01020 1F6F 4D             TST A
01025                     *  BRANCH IF LAST CHARACTER IN LINE
01030 1F70 27 0A          BEQ    ENDR
01040 1F72 81 19          CMP A  #$19      ANOTHER KEYWORD?
01050 1F74 2C 04          BGE    STRCHR    BRANCH IF NOT
01060 1F76 8D 13          BSR    STORKY    STORE KEYWORD IN BUFFER
01070 1F78 20 F3          BRA    SLOOP
01080 1F7A 8D 1C   STRCHR BSR    TRNSFR+2  STORE CHARACTER IN BUFFER
01090 1F7C DE 24   ENDR   LDX    BUFPNT    STORE $00 IN BUFFER
01100 1F7E A7 00          STA A  X
01110 1F80 CE 00B0        LDX    #BUFFER   PRINT CONTENTS OF THE BUFFER
01120 1F83 BD 0467        JSR    PSTR
01125                     *  SET INDEX TO LAST CHARACTER IN BUFFER
01130 1F86 DE 24          LDX    BUFPNT
01140 1F88 7E 1EBD        JMP    INPUT
01150 1F8B EE 00   STORKY LDX    X         STORE KEYWORD IN BUFFER
01160 1F8D 09             DEX
01170 1F8E 09      BACKUP DEX              LOCATE KEYWORD IN TABLE
01180 1F8F A6 00          LDA A  X
01190 1F91 26 FB          BNE    BACKUP
01195                     *  SKIP PAST JUMP ADDRESS IN TABLE
01200 1F93 08             INX
01210 1F94 08             INX
01220 1F95 08             INX
01230 1F96 A6 00   TRNSFR LDA A  X         TRANSFER CHARACTERS TO BUFFER
01240 1F98 81 20          CMP A  #32       LAST CHARACTER? OR KEY WORD?
01250 1F9A 2D 0E          BLT    ENDSTR    BRANCH IF SO
01260 1F9C 08             INX
01270 1F9D DF AC          STX    TEMP      STORE MEMORY POINTER
01280 1F9F DE 24          LDX    BUFPNT    PICK UP BUFFER INDEX
01290 1FA1 A7 00          STA A  X
01300 1FA3 08             INX
01310 1FA4 DF 24          STX    BUFPNT
01320 1FA6 DE AC          LDX    TEMP
01330 1FA8 20 EC          BRA    TRNSFR
01335                     * LOAD POSITION OF BUFFER AFTER LINE#
01340 1FAA DE 2C   ENDSTR LDX    MEMPNT
01345                     *  LOAD POSITION OF BUFFER AFTER LINE#
01350 1FAC 08             INX
01360 1FAD 08             INX
01370 1FAE 39             RTS
01380 1FAF 8C 00B0 ZERCSR CPX    #BUFFER   AT BEGINNING OF BUFFER?
01390 1FB2 27 08          BEQ    ENDZ      BRANCH IF YES
01400 1FB4 86 08          LDA A  #BACK
01410 1FB6 BD 0386        JSR    OUTC
01420 1FB9 09             DEX
01430 1FBA 20 F3          BRA    ZERCSR
01440 1FBC 39      ENDZ   RTS


01450 1FBD CE 00AF ENDRTN LDX    #BUFFER-1  FIND END OF LINE IN BUFFER
01460 1FC0 08      LOOP   INX
01470 1FC1 A6 00          LDA A  X
01480 1FC3 4D             TST A
01490 1FC4 26 FA          BNE    LOOP
01500 1FC6 7E 0453        JMP    PLINEC
01510        1FC9  PRGEND EQU    *
```

More

Control-D deletes the character at the current cursor position and moves all following characters one place to the left.

The preceding edit commands are summarized in Table 1.

Alternatively, typing any other character substitutes the new character instead of the old one, and moves the cursor right one place.

This recall-and-edit feature can be used to change any part of the line, even the line number. Changing the line number makes it into a new line, to be inserted into the program in line number order. The old line stays in the program, and the new line is added. Thus it is possible to produce identical (or slightly modified) copies of existing lines.

How It Works

Let's confine ourselves to describing the version 2.0 BASIC patch in Listing 1; the other versions work similarly.

The edit patch works by manipulating characters in the BASIC interpreter buffer that starts at location 00B0.

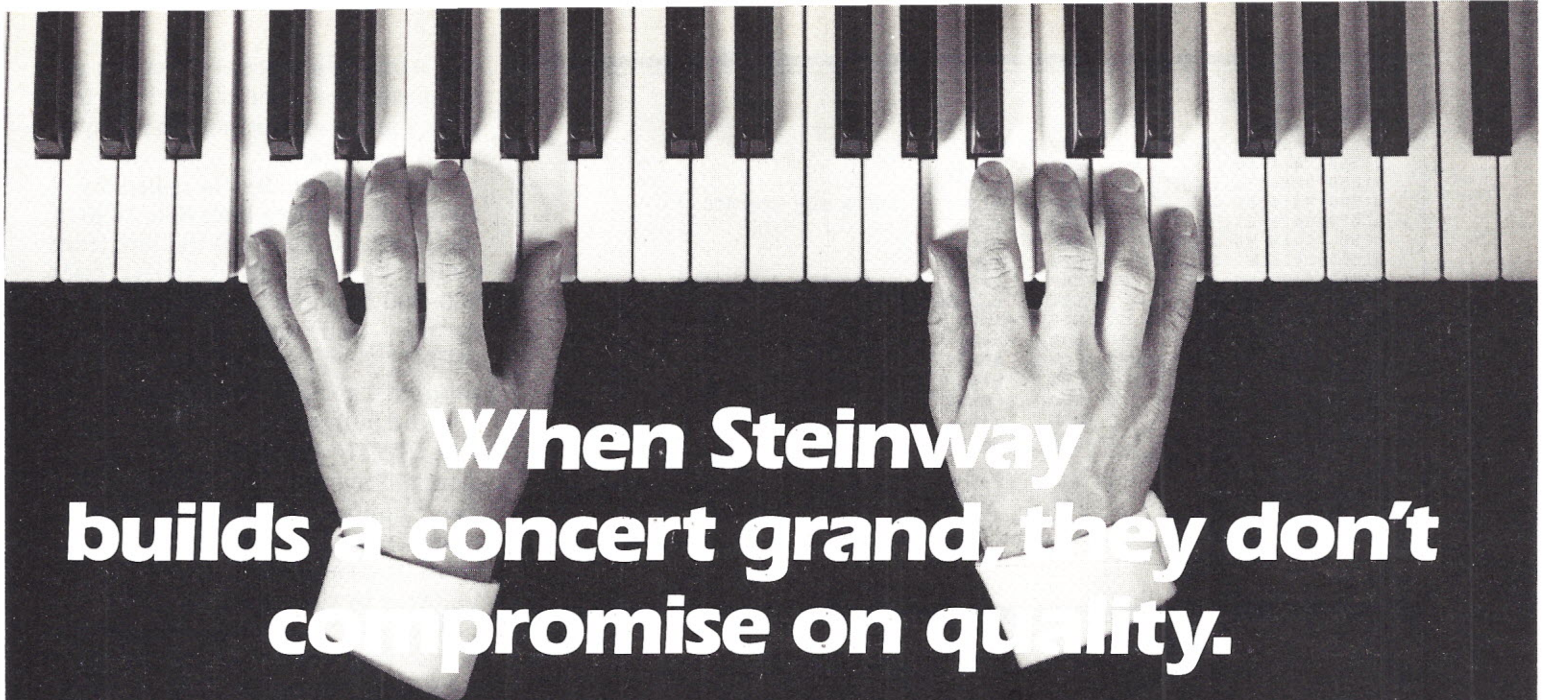
Lines 20–130 of the program define entry points to the BASIC interpreter. PSHX and PULX are two subroutines used to push and pull the index register into an index stack kept by BASIC. INPUTC and OUTC are the BASIC character input and output routines. NUMC is a BASIC routine which converts a number in ASCII in memory to a binary-coded decimal (BCD) number used by LINEC to find that line in memory.

PSTR is a routine which prints a string of characters. PLINEC is the entry point when a carriage return has been entered.

Lines 150–540 represent the edit input routine, which branches to the appropriate routines when the proper control characters are entered. It starts at 1EAF for BASIC version 2.0, 1EE2 for BASIC version 2.2 or 2698 for BASIC version 3.5.

The delete routine, called with a control-D, consists of lines 550–730. MOVELF shifts all characters in the buffer which are past the cursor position one place to the left. As it does this, it prints those characters on the CRT and counts them. MOVECS then puts the cursor back to where it started by printing that number of backspaces.

The add routine, called by a control-A and shown in lines 740–879, operates in a similar manner except



When Steinway builds a concert grand, they don't compromise on quality.

Neither do we.

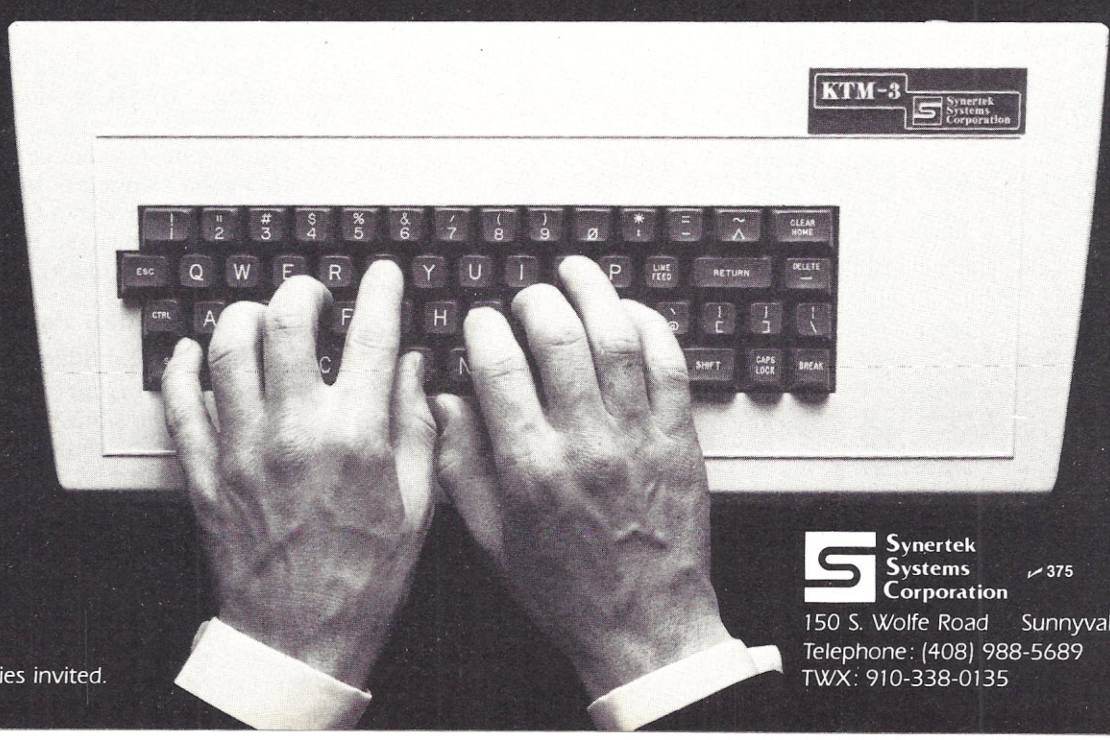
The low price is deceiving. Synertek Systems Corporation allows no compromise in the versatility, reliability, and quality of the KTM-3.

This complete, low-cost terminal offers no-frills, solid performance. With its power supply and connecting cables, the KTM-3 requires only a CRT monitor or television screen to provide a functioning system for data entry and display.

The KTM-3 incorporates all the latest features of standard terminals such as:

- Choice of character screen size:
  - 24×80 characters in the KTM-3/80
  - 24×40 characters in the KTM-3
- 110 to 19.2K baud
- 7×9 character matrix in an 8×10 field
- "CAPS LOCK" key
- 128 ASCII characters with descenders
- Full and half duplex with modem control
- Built-in power supply and cables included
- Scrolling and full cursor control
- Display of framing and parity errors
- Auto key-repeat
- Debug mode (Display of control characters)
- Built-in diagnostics
- 50/60 Hz operation with 220 volt version available

STARTING AT **$389**,
OUR PRICE WILL BE MUSIC TO YOUR EARS.



Synertek Systems Corporation ✓375
150 S. Wolfe Road Sunnyvale, CA 94086
Telephone: (408) 988-5689
TWX: 910-338-0135

Dealer inquiries invited.

Listing 1 continued.

```
01520 040F                ORG    $040F
01530 040F 81 05          CMP A  #5
01540 0411 26 03          BNE    CONT6
01550 0413 7E 1EAF        JMP    EDIT
01560 0416 B1 0153 CONT6  CMP A  $153      CHECK FOR CONTROL X
01570 0419 27 DE          BEQ    $3F9
01580 041B B1 0154        CMP A  $154
01590 041E 27 08          BEQ    CONT7
01600 0420 81 0D          CMP A  #$0D      CHECK FOR CAR RETURN
01610 0422 27 2F          BEQ    PLINEC
01620 0424 2D E6          BLT    $40C
01630 0426 20 0E          BRA    $436
01640 0428 B6 0155 CONT7  LDA A  $0155
01650 014E                ORG    $14E
01660 014E 1FC9           FDB    PRGEND
01665 0154                ORG    $154
01670 0154 08             FCB    BACK,0
      0155 00
01680 A048                ORG    $A048
01690 A048 0100           FDB    $100
```

Listing 2. Editing modification in SWTP Disk BASIC version 3.5.

```
                          NAM    EDITOR
                   *
                   * EDITOR FOR BASIC V3.5 AS SUPPLIED FOR DMAF1 DISK
                   *
                   * WRITTEN BY R.SILVER
                   *
                   * MODIFIED FOR THE DMAF1 DISK BASIC BY K.J.KROEKER
                   *
                   *
          0547     PSHX   EQU    $0547
          04F0     BERROR EQU    $04F0
          0428     INPUTC EQU    $428
          03F1     OUTC   EQU    $3F1
          055C     PULX   EQU    $055C
                          OPT    O,S
          0BA1     BERR2  EQU    $0BA1
          0BA4     NUMC   EQU    $0BA4
          0B61     LINEC  EQU    $0B61
          0520     PSTR   EQU    $0520
          050C     PLINEC EQU    $050C
     2698                 ORG    $2698
     2698 BD 0547  EDIT   JSR    PSHX
     269B 6F 00    CLEAR  CLR    ,X
     269D 08              INX
     269E 8C 00F8         CPX    #$00F8
     26A1 26 F8           BNE    CLEAR
     26A3 BD 055C         JSR    PULX
     26A6 BD 0428  INPUT  JSR    INPUTC
     26A9 81 0D           CMP A  #$0D
     26AB 26 03           BNE    CONT1
     26AD 7E 27A6         JMP    ENDRTN
     26B0 81 08    CONT1  CMP A  #$08
     26B2 27 2D           BEQ    BACKSPC
     26B4 81 09           CMP A  #$09
     26B6 27 15           BEQ    FORWARD
     26B8 81 04           CMP A  #$04
     26BA 27 34           BEQ    DELETE
     26BC 81 01           CMP A  #$01
     26BE 27 5D           BEQ    ADD
     26C0 81 12           CMP A  #$12
     26C2 26 03           BNE    CONT3
     26C4 7E 273B         JMP    RECALL
     26C7 81 1F    CONT3  CMP A  #$1F
     26C9 2B DB           BMI    INPUT
     26CB A7 00    STORCH STA A  ,X
     26CD A6 00    FORWAR LDA A  ,X
     26CF 26 07           BNE    CONT2
     26D1 86 08           LDA A  #$08
     26D3 BD 03F1         JSR    OUTC
     26D6 20 CE           BRA    INPUT
     26D8 08       CONT2  INX
     26D9 8C 00F8         CPX    #$00F8
     26DC 26 C8           BNE    INPUT
     26DE 7E 04F0  ERROR  JMP    BERROR
     26E1 8C 00B0  BACKSP CPX    #$00B0
     26E4 27 03           BEQ    ATBEG
     26E6 09              DEX
     26E7 20 BD    LOOPBA BRA    INPUT
     26E9 86 09    ATBEG  LDA A  #$09
     26EB BD 03F1         JSR    OUTC
     26EE 20 B6           BRA    INPUT
     26F0 BD 0547  DELETE JSR    PSHX
     26F3 7F 004A         CLR    $004A
     26F6 A6 01    MOVELF LDA A  1,X
     26F8 BD 03F1         JSR    OUTC
     26FB A7 00           STA A  X
```

More

that it inserts a space at the cursor position and then shifts all characters from the cursor position one character to the right.

The recall routine is called by a control-R and uses lines 880–1490 of the program. To recall the line from memory into the BASIC buffer, lines 890–980 locate the requested line in memory and store its position in MEMPNT. Lines 1010–1370 then transfer the characters from memory into the buffer.

In SWTP BASICs, each line is stored in memory as a two-byte BCD line number, a two-byte address which represents the encoded keyword (such as PRINT, INPUT or IF), followed by the remaining characters of the line. Before the line can be physically transferred from memory back into the buffer, this two-byte code must be translated back into the keyword; this is done by subroutine STORKY, which locates the keyword in the BASIC jump table and stores it in the buffer.

When editing is completed, lines 1450 through 1500 position the index at the end of the buffer and jump to the BASIC reentry point to process the carriage return and place the line back into memory in its correct location.

The patch to the BASIC input routine that calls the editor when a control-E is first entered is shown at lines 1520–1640. This patch is the same for both BASIC versions 2.0 and 2.2. BASIC version 3.5, however, uses a different input routine as well as different entry points. Listing 2 shows how Ken Kroeker of Dakota Collegiate Institute, Winnipeg, Canada, modified the patch to work with this BASIC.

Finally, lines 1650 through 1670 change BASIC's end-of-program pointer to leave room for the editor patch, and also change the backspace and echo characters so that the standard backspace (control-H or ASCII $08) is used instead of the standard control-O and underline echo.

TSC Text Editor Patch

The TSC Text Editor has a large variety of commands and operational modes, but it is line- rather than character-oriented. Moreover, it was originally intended for hard-copy as well as CRT terminal use. Thus, it is rather awkward to change characters in the middle of a line. The text editor patch described here can easily be implemented in the TSC editor to greatly

enhance its operation.

The original TSC Text Editor was cassette-based. It was later adapted, with relatively minor patches but without reassembly, to work on the SWTP disk system under MiniFlex, and on the Percom disk system under MINIDOS. Since all three versions resulted from the same assembly, Listing 4 applies to all of them except for the end-of-program pointers.

This affects the two ORG statements labelled as ***SEE TEXT*** in Listing 4. The addresses shown ($1D7D and $16DB) apply to Percom's "Touchup" version of the Editor.

In the original cassette version, the first ORG should read ORD $1492. The last ORG, as well as the LDX #PRGEND following it, should be omitted.

In the MiniFlex version, the first ORG should read $19DB, and the last ORG and the following LDX should also be omitted.

A/BASIC Modifications

Of all the 6800 higher-level language compilers, A/BASIC is the only one available in a cassette version. Since some recent statistics seem to indicate that there are more tape users than was thought, it's worth some space and time.

Though A/BASIC is relatively inexpensive ($65 in the cassette version), it isn't nearly as popular as one would expect. There are several reasons.

First, Microware's ads state that the cassette version requires the RT/68 operating system. Though RT/68 is quite impressive as a multitasking, real-time execution program, it is not nearly as versatile as some other, more plain monitors. Nor is it compatible with any of them. (In fact, even if a program is compiled with unmodified A/BASIC on an RT/68 system, it will not run on computers with other monitors.) This, combined with its price of $55, has not made it popular for the nonprofessional. (Microware, 5835 Grand Ave., Des Moines, IA 50304) has sold many more RT/68 manuals—to colleges and universities for text use—than ROMs.)

Second, the cassette version of A/BASIC is awkward to use. In fact, any compiler is awkward to use, but in a cassette form it is downright painful.

To run A/BASIC from cassette normally requires the following steps:

Listing 2 continued.

```
26FD 08                 INX
26FE 7C 004A            INC     $004A
2701 4D                 TST A
2702 26 F2              BNE     MOVELFT
2704 86 20      SPACE   LDA A   #$20
2706 BD 03F1            JSR     OUTC
2709 7D 004A    MOVECS  TST     $4A
270C 27 0A              BEQ     ENDD
270E 86 08              LDA A   #$08
2710 BD 03F1            JSR     OUTC
2713 7A 004A            DEC     $4A
2716 20 F1              BRA     MOVECSR
2718 BD 055C    ENDD    JSR     PULX
271B 20 CA              BRA     LOOPBACK
271D BD 0547    ADD     JSR     PSHX
2720 7F 004A            CLR     $4A
2723 C6 20              LDA B   #$20
2725 17         MOVERT  TBA
2726 E6 00              LDA B   X
2728 A7 00              STA A   X
272A BD 03F1            JSR     OUTC
272D 8C 00FB            CPX     #$FB
2730 08                 INX
2731 27 AB              BEQ     ERROR
2733 7C 004A            INC     $4A
2736 4D                 TST A
2737 26 EC              BNE     MOVERT
2739 20 C9              BRA     SPACE
273B 8D 5B      RECALL  BSR     ZERCSR
273D CE 00B0            LDX     #$00B0
2740 BD 0BA4            JSR     NUMC
2743 24 03              BCC     CONT4
2745 7E 0BA1    ERR2    JMP     BERR2
2748 DF 24      CONT4   STX     $24
274A BD 0B61            JSR     LINEC
274D 25 F6              BCS     ERR2
274F 08                 INX
2750 08                 INX
2751 DF 2C              STX     $2C
2753 8D 1F              BSR     STORKYWD
2755 08                 INX
2756 A6 00      SLOOP   LDA A   X

2758 4D                 TST A
2759 27 0A              BEQ     ENDR
275B 81 19              CMP A   #$19
275D 2C 04              BGE     STRCHR
275F 8D 13              BSR     STORKYWD
2761 20 F3              BRA     SLOOP
2763 8D 1C      STRCHR  BSR     TRNSFR+2
2765 DE 24      ENDR    LDX     $24
2767 A7 00              STA A   X
2769 CE 00B0            LDX     #$B0
276C BD 0520            JSR     PSTR
276F DE 24              LDX     $24
2771 7E 26A6            JMP     INPUT
2774 EE 00      STORKY  LDX     X
2776 09                 DEX
2777 09         BACKUP  DEX
2778 A6 00              LDA A   X
277A 26 FB              BNE     BACKUP
277C 08                 INX
277D 08                 INX
277E 08                 INX
277F A6 00      TRNSFR  LDA A   X
2781 81 20              CMP A   #32
2783 2D 0E              BLT     ENDSTR
2785 08                 INX
2786 DF AC              STX     $AC
2788 DE 24              LDX     $24
278A A7 00              STA A   X
278C 08                 INX
278D DF 24              STX     $24
278F DE AC              LDX     $AC
2791 20 EC              BRA     TRNSFR
2793 DE 2C      ENDSTR  LDX     $2C
2795 08                 INX
2796 08                 INX
2797 39                 RTS
2798 8C 00B0    ZERCSR  CPX     #$B0
279B 27 08              BEQ     ENDZ
279D 86 08              LDA A   #$08
279F BD 03F1            JSR     OUTC
27A2 09                 DEX
27A3 20 F3              BRA     ZERCSR
27A5 39         ENDZ    RTS
27A6 CE 00AF    ENDRTN  LDX     #$AF
27A9 08         LOOP    INX
27AA A6 00              LDA A   X
27AC 4D                 TST A
27AD 26 FA              BNE     LOOP
27AF 7E 050C            JMP     PLINEC
27B2 81 05      PATCH3  CMP A   #5
27B4 26 03              BNE     GOBACK
27B6 7E 2698            JMP     EDIT
27B9 B1 AC00    GOBACK  CMP A   $AC00
27BC 7E 04CA            JMP     $4CA
```

More

Listing 2 continued.

```
      27BF    PRGEND EQU    *
04C7                 ORG    $4C7
04C7 7E 27B2         JMP    PATCH3
014E                 ORG    $14E
014E 27BF            FDB    PRGEND
A048                 ORG    $A048
A048 0100            FDB    $100
                     END
PSHX   0547
BERROR 04F0
INPUTC 0428
OUTC   03F1
PULX   055C
BERR2  0BA1
NUMC   0BA4
LINEC  0B61
PSTR   0520
PLINEC 050C
EDIT   2698
CLEAR  269B
INPUT  26A6
CONT1  26B0
CONT3  26C7
STORCH 26CB
FORWAR 26CD
CONT2  26D8
ERROR  26DE
BACKSP 26E1
LOOPBA 26E7
ATBEG  26E9
DELETE 26F0
MOVELF 26F6
SPACE  2704
MOVECS 2709
ENDD   2718
ADD    271D
MOVERT 2725
RECALL 273B
ERR2   2745
CONT4  2748
SLOOP  2756
STRCHR 2763
ENDR   2765
STORKY 2774
BACKUP 2777
TRNSFR 277F
ENDSTR 2793
ZERCSR 2798
ENDZ   27A5
ENDRTN 27A6
LOOP   27A9
PATCH3 27B2
GOBACK 27B9
PRGEND 27BF

TOTAL ERRORS 00000
```

Listing 3. Editing modification for Percom Super BASIC.

```
               NAM    EDITOR    FOR SUPER BASIC
        *             R.SILVER MAY 13,1979
        * BASIC ENTRY POINTS
(08B5)  PSHX   EQU    $08B5     PUSH INDEX
(0081)  BUFFER EQU    $81       BEGINNING OF THE BASIC BUFFER
(00C9)  BUFEND EQU    $C9       END OF BUFFER
(10C7)  BERROR EQU    $10C7     ERROR ROUTINE
(0570)  INPUTC EQU    $0570     INPUT ROUTINE
(0561)  OUTC   EQU    $0561     OUTPUT ROUTINE
(08CA)  PULX   EQU    $08CA     PULL INDEX
(002A)  COUNTR EQU    $2A
(004A)  BUFPNT EQU    $4A       POINTER TO BUFFER POSITION
(002E)  MEMPNT EQU    $2E       POINTER TO MEMORY POSITION
(0047)  TEMP   EQU    $47
(10C7)  BERR2  EQU    $10C7     ERROR ROUTINE
(1414)  NUMC   EQU    $1414     NUMERIC CONVERSION
(0F22)  LINEC  EQU    $0F22     FIND LINE IN MEMORY
(08A4)  PSTR   EQU    $08A4     PRINT STRING ROUTINE
(0885)  PLINEC EQU    $0885     BRANCH POINT BACK TO BASIC
        * MISC CONSTANTS
(0008)  BACK   EQU    $08       BACK SPACE CHARACTER
(0009)  TAB    EQU    $09       FORWARD TAB CHARACTER
(3000)         ORG    $3000
```

More

1. Load Microware's RTEDIT editor (supplied on the A/BASIC cassette) into the computer.

2. Run RTEDIT to enter and edit the source code.

3. When done, save the source code to cassette.

4. Load the A/BASIC compiler into memory.

5. Set up two cassette recorders—one to read the source code, the other to record the machine-language code. The recorder used to read the source code must have motor control so that A/BASIC can start and stop it as needed.

6. Start A/BASIC and start reading the source code.

7. A/BASIC is a two-pass compiler which must read the source code twice. After the first read is finished, rewind the tape.

8. Start both tape recorders and continue A/BASIC.

9. When the compiler is done, rewind both tapes and use the monitor's L function to read the machine language into the computer.

10. Now execute the program.

11. Since programs never run the first time, figure out what went wrong and go back to step 1 to start all over again, as many times as are needed to make it work.

Sound tedious? Right! Add to this the necessity to get RT/68, two tape recorders and motor control, and the process becomes expensive as well.

But there is a way out. With a little patching it is possible to use A/BASIC without spending as much time and without all the extra hardware. The patches do two things—allow RTEDIT and A/BASIC to work with MIKBUG or SWTBUG, and keep the source code in memory so it does not have to be read twice from tape by the compiler.

Listing 5 shows the patches required to RTEDIT. The first portion (down to location 01A6) substitutes the MIKBUG addresses for INEEE and OUTEEE instead of the addresses used by RT/68.

Immediately following is a short subroutine, at A04A, for printing a carriage return and line feed. RT/68 contains such a CR/LF subroutine, and Microware uses it both in the editor as well as in A/BASIC and even compiled programs. Thus, we need a short subroutine in RAM.

This first portion of the patch is all you need if you are using the SWTP AC-30 cassette interface with motor control and two recorders. However,

16 BIT 8086 MICRO SYSTEM

CP/M -86
TEC -86
BASIC -86

$3990

Available Now Basic-86™-$350 CP/M-86™-$250

- IEEE 696 S-100 • CPU w/Vectored Interrupts • 24 Parallel I/O lines • Two RS-232 Ports • Baud rates from 50 to 19200 • Independent baud rate for each serial port • 32K static RAM (Expandable) • ROM boot for CP/M-86™ • Dual 8-inch Shugart floppies with controller • Attractive desk top enclosure • 64K Dynamic RAM opt. - Available Soon

Apple A/D Board	16 channel 12 bit	$495
Apple D/A and Parallel Output Board	2 channel 12 bit	$295
S-100 A/D Timer - TOD - Counter Board	16 channel 12 bit	$695
S-100 D/A Board	4 channel 12 bit	$395

Call for complete product line information.

TECMAR INC.

TECMAR, INC. 139
(216)382-7599
464-7410
23600 Mercantile Rd. • Cleveland, OH 44122

RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILITIES

HARD DISK MULTIPLEXOR WITH 10 MEG HARD DISK
FOR THE TRS-80* Mod II

NOW YOU CAN HAVE THAT LARGE COMMON DATA BASE!!

- Allows up to 4 Mod II's to connect to a single controller — up to 4 hard disk drives per controller. Users may access the same file simultaneously (first-come first-served).
- Uses Cameo controller and standard 10-megabyte cartridge (hard) disk drives along with RACET Hard/Soft Disk System (HSD) software.
- Access times 3 to 8 times faster than floppy. Mixed floppy/hard disk operation supported.
- Compatible with your existing TRSDOS programs! All BASIC statements are identical.
- A single file may be as large as one disk. Alternate mode allows 24-million byte record range. Directory expandable to handle thousands of files.
- Includes special utilities — backup and copies. HPURGE for multiple deletions. HDCS directory catalog system. and Hard Disk Superzap. FORMAT utility includes options for specifying sectors/gran. platters/drive. logical disk size. etc.

HARD DISK DRIVE & CONTROLLER $5995 **RACET HSD Software $400**
Call for multiuser pricing. Dealers call for OEM pricing.

BASIC LINK FACILITY 'BLINK' **$25 Mod I, $50 Mod II**
Link from one BASIC program to another saving all variables! The new program can be smaller or larger than the original program in memory. The chained program may either replace the original program, or can be **merged** by statement number. The statement number where the chained program execution is to begin may be specified! (Mod I Min 32K 1-disk)

INFINITE BASIC (Mod I Tape or Disk) **$49.95**
Extends Level II BASIC with complete MATRIX functions and 50 more string functions. Includes RACET machine language sorts! Sort 1000 elements in 9 seconds!! Select only functions you want to optimize memory usage.

INFINITE BUSINESS (Requires Infinite BASIC) **$29.95**
Complete printer pagination controls — auto headers, footers, page numbers. Packed decimal arithmetic — 127 digit accuracy +, −, *, /. Binary search of sorted and unsorted arrays. Hash codes.

*TRS-80 IS A REGISTERED TRADEMARK OF TANDY CORPORATION

CHECK, VISA, M/C, C.O.D., PURCHASE ORDER
TELEPHONE ORDERS ACCEPTED (714) 637-5016

101
RACET COMPUTES
702 Palmdale, Orange, CA 92665

RACET SORTS — RACET UTILITIES — RACET computes — RACET SORTS — RACET UTILIT

FMG CORPORATION 70

Send for FREE CATALOG!

THE PASCAL/MT As Low As $250.00

- Compiler executes under the CP/M operating system in as little as 32 K bytes of RAM
- Interactive Symbolic Debugger which enables the programmer to examine variables, set a breakpoint, and trace procedure calls interactively at run time
- Compiles at the rate of 600 lines per minute on a 2 MHZ 8080
- Programs Execute up to 10 TIMES FASTER than popular interpretive Pascals
- The code generated is 8080 object code which is ROMable with a minimum run time overhead of 1.5K bytes
- Interrupt procedures allow the programmer to write interrupt drivers for I/O and other real time tasks in Pascal/MT
- Bit manipulations of variables may be performed with the built-in procedures: SETBIT, CLRBIT, TSTBIT, SHL, SHR, SWAP, LO, HI.
- Assembly language subroutines may be called from Pascal/MT
- Business arithmetic version of Pascal/MT is also available
- Pascal data structures supported are: ENUMERATION AND SUBRANGE TYPES, RECORD, ARRAY, REAL, INTEGER, CHAR, and BOOLEAN
- Not implemented are: SETS, GOTO, GET, PUT

VISA M-451 master charge

BUSINESS CONTROL PROGRAMS
FROM THE ORIGINATOR OF THE TRS-80® PROJECT

FMG CORPORATION
P.O. Box 16020
Ft. Worth, TX
76133
(817) 294-2510

CP/M®2

- Enhanced Upward Compatible File System
- Powerful New Random Access Capabilities

FMG Corporation now offers the CP/M 2.2 for the TRS-80 Model II. From minidisks, floppy disks, all the way to high-capacity hard disks, the flexibility of CP/M 2.2 makes it a truly universal operating system. The package includes an 8" system disk, editor, assembler and debugger for the TRS-80 Model II.

AS LOW AS $200.00

BUSINESS

- General Ledger
- Payroll
- Accounts Payable
- Accounts Receivable

These business systems are designed with the business manager in mind! Totally screen oriented, they give complete user prompting as each entry is required. Major changes in your current bookkeeping method are not necessary to make these programs work for you.

AS LOW AS $500.00

CP/M is a registered trademark of Digital Research Corp. TRS-80 is a registered trademark of Radio Shack

Listing 3 continued.

```
3000 BD 08B5  EDIT   JSR    PSHX
3003 6F 00    CLEAR  CLR    ,X         CLEAR BUFFER TO END
3005 08              INX
3006 8C 00C9         CPX    #BUFEND    IS IT THE END OF THE BUFFER?
3009 26 F8           BNE    CLEAR
300B BD 08CA         JSR    PULX
300E BD 0570  INPUT  JSR    INPUTC     INPUT LOOP
3011 81 0D           CMP A  #$0D       CHECK FOR CARRIAGE CONTROL
3013 26 03           BNE    CONT1
3015 7E 310E         JMP    ENDRTN     IF SO JUMP BACK TO BASIC
3018 81 08    CONT1  CMP A  #BACK      CHECK FOR BACKSPACE
301A 27 2D           BEQ    BACKSP
301C 81 09           CMP A  #TAB
301E 27 15           BEQ    FORWRD
3020 81 04           CMP A  #$04       CONTROL D
3022 27 34           BEQ    DELETE
3024 81 01           CMP A  #$01       CONTROL A
3026 27 5D           BEQ    ADD
3028 81 12           CMP A  #$12       CONTROL R
302A 26 03           BNE    CONT3
302C 7E 30A3         JMP    RECALL
302F 81 1F    CONT3  CMP A  #$1F       CHECK IF CONTROL CHARACTER
3031 2B DB           BMI    INPUT      IF SO IGNORE
3033 A7 00    STORCH STA A  ,X         STORE CHARACTER IN BUFFER
              * MOVE CURSOR FORWARD
3035 A6 00    FORWRD LDA A  ,X
              * CHECK IF AT END OF LINE IN BUFFER
3037 26 07           BNE    CONT2      BRANCH IF NOT
3039 86 08           LDA A  #BACK
              * PUT CURSOR BACK WHERE IT BELONGS
303B BD 0561         JSR    OUTC
303E 20 CE           BRA    INPUT
              * MOVE TO NEXT POSITION IN BUFFER
3040 08       CONT2  INX               MOVE TO NEXT POSITION IN BUFFER
3041 8C 00C9         CPX    #BUFEND    CHECK FOR END OF BUFFER
3044 26 C8           BNE    INPUT
3046 7E 10C7  ERROR  JMP    BERROR
              * BACKUP CURSOR
3049 8C 0081  BACKSP CPX    #BUFFER    CHECK IF AT START OF BUFFER
304C 27 03           BEQ    ATBEG      BRA IF SO
304E 09              DEX
304F 20 BD    LOOPBA BRA    INPUT
3051 86 09    ATBEG  LDA A  #TAB       MOVE CURSOR FORWARD
3053 BD 0561         JSR    OUTC
3056 20 B6           BRA    INPUT


              * DELETE A CHARACTER FROM BUFFER AND CRT
3058 BD 08B5  DELETE JSR    PSHX
              * ZERO COUNTER IN ORDER TO REPOSITION CURSOR
305B 7F 002A         CLR    COUNTR
              * MOVE CURSOR AND BUFFER CONTENTS ONE POSITION LEFT
305E A6 01    MOVLFT LDA A  1,X
3060 BD 0561         JSR    OUTC
3063 A7 00           STA A  X
3065 08              INX
3066 7C 002A         INC    COUNTR     COUNT CHARACTERS TO END
3069 4D              TST A
306A 26 F2           BNE    MOVLFT
306C 86 20    SPACE  LDA A  #$20       PRINT SPACE
306E BD 0561         JSR    OUTC
              * PUT CURSOR BACK TO CORRECT POSITION
3071 7D 002A  MOVECS TST    COUNTR     CHECK TO SEE IF FINISHED
3074 27 0A           BEQ    ENDD
3076 86 08           LDA A  #BACK
3078 BD 0561         JSR    OUTC
307B 7A 002A         DEC    COUNTR
307E 20 F1           BRA    MOVECS
3080 BD 08CA  ENDD   JSR    PULX
3083 20 CA           BRA    LOOPBA


3085 BD 08B5  ADD    JSR    PSHX
              * ADD SPACE TO BUFFER AT CURSOR POSITION
3088 7F 002A         CLR    COUNTR     ZERO COUNTER
```

More

the rest of the patch will allow the source code to stay in memory while the compiler is loaded. (You'd still want to make a cassette tape of the edited source code now and then to guard against program loss, but this tape would be strictly for backup and not needed by the compiler.)

Listing 6 shows the patches required to the A/BASIC compiler itself. As before, the first part of Listing 6 must always be done to use A/BASIC with non-RT/68 systems. It patches A/BASIC to work with MIKBUG/SWTBUG-compatible monitors. Again, we see a CR/LF routine at A04A.

If you are not using two recorders and motor control, then the second half of the patch is required as well. This part allows A/BASIC to use the source code which was left in memory by RTEDIT. Note that A/BASIC needs to read the source code twice, but since the code is already in memory, that is easy to arrange without any extra work.

A/BASIC still needs a cassette recorder to store the machine-language object code, but this one need not have motor control. A/BASIC outputs the machine code fairly fast, but still slowly enough that the monitor will have no trouble reading it later. (It comes out in the standard S1....MIKBUG format.)

Once the machine-language code is on cassette, there is one more thing to do. A/BASIC does not insert CR/LF codes into the object code, but normally calls the appropriate routine in RT/68. The code output by the modified compiler will now call the routine at A04A instead. This means that this routine must be inserted into memory any time the object program is run.

If the program is to be run often, or if you intend to give a copy to someone else, it is much more convenient to append a short CR/LF routine to the end of the compiled program (the A/BASIC compiler tells you where the end is) and change all the references from A04A to the new address. Use a search program to find all occurrences of A04A in the object code and replace them with the address of the added subroutine.

A/BASIC Disk Operation

A/BASIC is, of course, most convenient to use in a disk system. Microware sells a disk version of A/BASIC for SWTP and SSB disk systems, but it costs $150 rather than

$65 for the cassette version.

If you don't need the disk files which the disk version allows, then you can patch the cassette version to work with a disk. A patch for the Percom disk system is part of the Percom Users' Library, and the patch for miniFlex is shown in Listing 7.

This program patches the cassette compiler to work with MIKBUG/SWTBUG systems, and to allow the use of miniFlex text files. The source code can be generated with either the BUILD command or, better yet, with the Text Editor.

Rather than write the compiled object code back on disk, this patch leaves it in memory starting at location $2000 and returns to the monitor when done.

A/BASIC source code specifies by means of an ORG statement the location where the object code should go. Regardless of what origin is specified, this patch will place the object code starting at $2000. If the code was origined at $2000, then it is in the right place and can be executed immediately after compilation, or saved with the disk operating system's SAVE command for later use.

If the program was origined for a different location, then it must be moved there before it can be run. A memory move utility is part of the TSC Flex Utilities package, or else the MOVE routine (described in the September 1980 installment) can also be used.

Both the A/BASIC compiler as well as the compiled object program use Flex's PCRLF routine at $711E for carriage returns and line feeds. If the compiled program is to be run on a non-miniFlex system, then it will again be necessary to insert a CR/LF routine somewhere, search out all the jumps to $711E and substitute jumps to the new routine.

Listing 7 is set up for a 32K or larger system. If it is run on a smaller system, the instruction at 16BA will have to be changed to reflect the lower memory limit. It is presently set at $6FFF so that the compiled program has 20K of room from $2000 to $6FFF, but does not erase miniFlex at $7000.

One last thought—although it is tempting to try disassembling a compiler such as A/BASIC, and then reassembling on a 6809 system to produce a 6809 compiler, don't bother. Even if it runs, it will still generate 6800 code! Changing it enough to produce 6809 output is a tough job. ■

Listing 3 continued.

```
308B C6 20           LDA B #$20       PUT SPACE IN BUFFER
                * MOVE CURSOR AND BUFFER RIGHT ONE PLACE
308D 17         MOVERT TBA
308E E6 00           LDA B X
3090 A7 00           STA A X
3092 BD 0561         JSR   OUTC
3095 08              INX
3096 8C 00C9         CPX   #BUFEND   CHECK FOR FILLED BUFFER
3099 27 AB           BEQ   ERROR
309B 7C 002A         INC   COUNTR    COUNT CHARACTERS FILLED
309E 4D              TST A
309F 26 EC           BNE   MOVERT
30A1 20 C9           BRA   SPACE
                * RECALL LINE FROM MEMORY AND PLACE IN BUFFER
30A3 8D 5B      RECALL BSR   ZERCSR    MOVE CURSOR TO START OF LINE
30A5 CE 0081         LDX   #BUFFER   FIND LINE NUMBER IN BUFFER
30A8 BD 1414         JSR   NUMC      CONVERT TO BCD FOR LINEC
30AB 24 03           BCC   CONT4     INVALID LINE NUMBER?
30AD 7E 10C7    ERR2 JMP   BERR2
                * STORE POSITION OF END OF LINE NUMBER
30B0 DF 4A      CONT4 STX  BUFPNT
30B2 BD 0F22         JSR   LINEC     FIND LINE IN MEMORY
30B5 25 F6           BCS   ERR2      BRANCH IF NOT FOUND
                * SKIP PAST LINE NUMBER STORED IN MEMORY
30B7 08              INX
30B8 08              INX
                * STORE POSITION OF LINE IN MEMORY
30B9 DF 2E           STX   MEMPNT
30BB 8D 1F           BSR   STORKY    STORE KEYWORD IN BUFFER
30BD 08              INX
                * TRANSFER CHARACTERS FROM MEMORY TO BUFFER
30BE A6 00      SLOOP LDA A X
30C0 4D              TST A
                * BRANCH IF LAST CHARACTER IN LINE
30C1 27 0A           BEQ   ENDR
30C3 81 19           CMP A #$19      ANOTHER KEYWORD?
30C5 2C 04           BGE   STRCHR    BRANCH IF NOT
30C7 8D 13           BSR   STORKY    STORE KEYWORD IN BUFFER
30C9 20 F3           BRA   SLOOP
30CB 8D 1C      STRCHR BSR   TRNSFR+2  STORE CHARACTER IN BUFFER
30CD DE 4A      ENDR LDX   BUFPNT    STORE $00 IN BUFFER
30CF A7 00           STA A X
30D1 CE 0081         LDX   #BUFFER   PRINT CONTENTS OF THE BUFFER
30D4 BD 08A4         JSR   PSTR
                * SET INDEX TO LAST CHARACTER IN BUFFER
30D7 DE 4A           LDX   BUFPNT
30D9 7E 300E         JMP   INPUT
30DC EE 00      STORKY LDX   0,X       STORE KEYWORD IN BUFFER
30DE 09              DEX
30DF 09         BACKUP DEX             LOCATE KEYWORD IN TABLE
30E0 A6 00           LDA A X
30E2 26 FB           BNE   BACKUP
                * SKIP PAST JUMP ADDRESS IN TABLE
30E4 08              INX
30E5 08              INX
30E6 08              INX
30E7 A6 00      TRNSFR LDA A 0,X       TRANSFER CHARACTERS TO BUFFER
30E9 81 20           CMP A #32       LAST CHARACTER? OR KEY WORD?
30EB 2D 0E           BLT   ENDSTR    BRANCH IF SO
30ED 08              INX
30EE DF 47           STX   TEMP      STORE MEMORY POINTER
30F0 DE 4A           LDX   BUFPNT    PICK UP BUFFER INDEX
30F2 A7 00           STA A X
30F4 08              INX
30F5 DF 4A           STX   BUFPNT
30F7 DE 47           LDX   TEMP
30F9 20 EC           BRA   TRNSFR
                * LOAD POSITION OF BUFFER AFTER LINE#
30FB DE 2E      ENDSTR LDX   MEMPNT
                * LOAD POSITION OF BUFFER AFTER LINE#
30FD 08              INX
30FE 08              INX
30FF 39              RTS
3100 8C 0081    ZERCSR CPX   #BUFFER   AT BEGINNING OF BUFFER?
3103 27 08           BEQ   ENDZ      BRANCH IF YES
3105 86 08           LDA A #BACK
3107 BD 0561         JSR   OUTC
310A 09              DEX
```

More

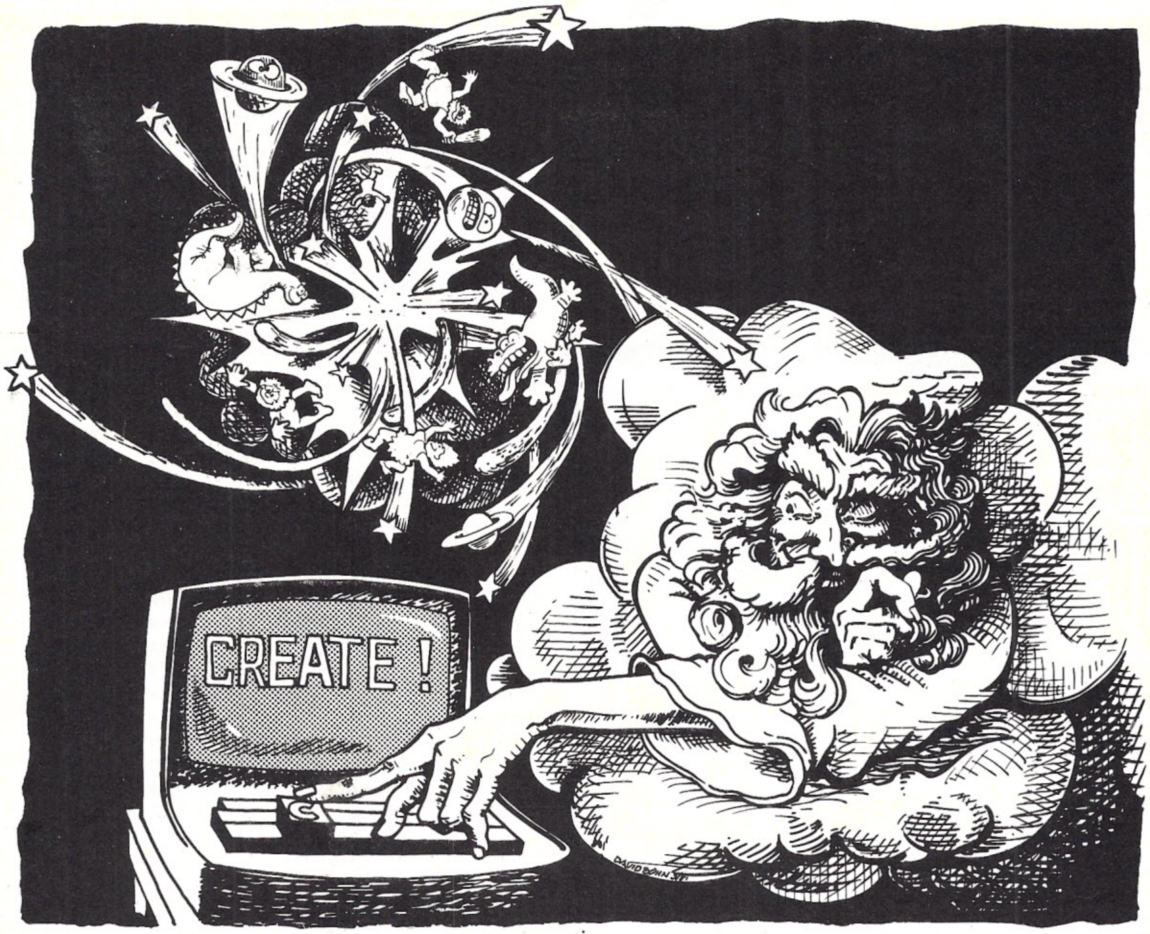
Listing 3 continued.

```
310B 20 F3            BRA    ZERCSR
310D 39        ENDZ   RTS


310E CE 0080   ENDRTN LDX    #BUFFER-1 FIND END OF LINE IN BUFFER
3111 08        LOOP   INX
3112 A6 00            LDA A X
3114 4D               TST A
3115 26 FA            BNE    LOOP
3117 7E 0885          JMP    PLINEC
   (311A)      PRGEND EQU    *


   (0847)             ORG    $0847
0847 81 05            CMP A #5
0849 26 03            BNE    CONT6
084B 7E 3000          JMP    EDIT
084E B1 0153   CONT6  CMP A $153         CHECK FOR CONTROL X
0851 27 D4            BEQ    $827
0853 B1 0154          CMP A $154
0856 27 09            BEQ    CONT7
0858 81 0D            CMP A #$0D         CHECK FOR CAR RETURN
085A 27 29            BEQ    PLINEC
085C 2D E3            BLT    $841
085E 20 0F            BRA    $86F
0860 01               NOP
0861 B6 0155   CONT7  LDA A $0155
   (014E)             ORG    $14E
014E 31 1A            FDB    PRGEND
```

Listing 4. Editing modification for the TSC Text Editor.

```
                      NAM    TSCEDIT   CURSOR EDIT FOR TSC EDITOR
               * R. SILVER APR 17,1979
   (04BB)      BUFLIM EQU    $4BB
   (0040)      TEMP   EQU    $40
   (008E)      CHRCNT EQU    $8E
   (0099)      FILEND EQU    $99
   (0097)      FILBEG EQU    $97
   (0091)      NUMBER EQU    $91
   (0206)      INPUTC EQU    $206
   (073B)      CLASS  EQU    $73B
   (0209)      OUTC   EQU    $209
   (038A)      TEDIT  EQU    $38A
   (00BB)      BUFFER EQU    $BB
   (045F)      BERR2  EQU    $45F
   (0755)      NUMC   EQU    $755
   (07AB)      LINEC  EQU    $7AB
   (0485)      PSTR   EQU    $485
   (03CC)      PLINEC EQU    $3CC

   (1D7D)             ORG    $1D7D     *** SEE TEXT ***
1D7D BD 0206   PATCH  JSR    INPUTC
1D80 81 05            CMP A #5         IS IT CONTROL E?
1D82 27 03            BEQ    EDIT
1D84 7E 049C          JMP    INCHR1
1D87 DF 40     EDIT   STX    TEMP
1D89 6F 00     CLEAR  CLR    ,X
1D8B 08               INX
1D8C 8C 0143          CPX    #BUFFER+136
1D8F 26 F8            BNE    CLEAR
1D91 DE 40            LDX    TEMP
1D93 BD 0206   INPUT  JSR    INPUTC
1D96 81 0D            CMP A #$0D
1D98 26 03            BNE    CONT1
1D9A 7E 1E77          JMP    ENDRTN
1D9D 81 08     CONT1  CMP A #$08
1D9F 27 29            BEQ    BACKSP
1DA1 81 09            CMP A #$09
1DA3 27 15            BEQ    FORWRD
```

More



if He'd used select[tm] it wouldn't have taken seven days

Learn SELECT* in just 90 minutes. A whole new word processing software concept that kicks the coded key habit and frees you from complicated instruction manuals. *SELECT* is fast. *SELECT* is logical. With single key mnemonics, you'll use dozens of commands that instantly access the rich capabilities of this system. There's nothing like it.

Simply hit "C" and you'll be ready to Create a document.
Key "I" and you'll be in the Insert mode.
Key "M" and Move entire blocks of text . . . and key dozens more.

That's all there is to it. You'll get all that word processing software promises . . . plus a few surprises.

SELECT with SUPERSPELL.* The only microcomputer software with an integrated spelling dictionary. To proof your text all you do, of course, is to key "S". *SUPERSPELL* with its 10,000 word dictionary scans your text at computer speed then displays and corrects all your typing errors. You can increase *SUPERSPELL*'s word power and customize the dictionary by adding new words, one at a time. Ask to see it today at your local dealer.

SELECT with SUPERSPELL . . . just a little byte more.™

SELECT will run on any machine that uses CP/M or MP/M** or its derivatives. It needs 40K of RAM and two disk drives. Special version now available for Radio Shack Mod II*** and Apple II****

* SELECT and SUPERSPELL are trademarks of Select Information Systems Inc.
** CP/M and MP/M are trademarks of Digital Research
*** A trademark of Tandy Corp.
**** A trademark of Apple Computer Inc.

✓374

select INFORMATION SYSTEMS™ 919 Sir Francis Drake Boulevard • Kentfield, California 94904 • (415) 459-4003

Listing 4 continued.

```
1DA5 81 04          CMP A #$04
1DA7 27 30          BEQ   DELETE
1DA9 81 01          CMP A #$01
1DAB 27 59          BEQ   ADD
1DAD 81 12          CMP A #$12
1DAF 26 03          BNE   CONT3
1DB1 7E 1E23        JMP   RECALL
1DB4 81 1F   CONT3  CMP A #$1F
1DB6 2B DB          BMI   INPUT
1DB8 A7 00   STORCH STA A ,X
1DBA A6 00   FORWRD LDA A ,X
1DBC 26 07          BNE   CONT2
1DBE 86 08          LDA A #$08
1DC0 BD 0209        JSR   OUTC
1DC3 20 CE          BRA   INPUT
1DC5 BD 04BB CONT2  JSR   BUFLIM
1DC8 20 C9          BRA   INPUT
1DCA 8C 00BB BACKSP CPX   #BUFFER
1DCD 27 03          BEQ   ATBEG
1DCF 09             DEX
1DD0 20 C1   LOOPBA BRA   INPUT
1DD2 86 09   ATBEG  LDA A #$09
1DD4 BD 0209        JSR   OUTC
1DD7 20 BA          BRA   INPUT
1DD9 DF 40   DELETE STX   TEMP
1DDB 7F 0091        CLR   NUMBER
1DDE A6 01   MOVLFT LDA A 1,X
1DE0 16             TAB
1DE1 BD 0209        JSR   OUTC
1DE4 17             TBA
1DE5 A7 00          STA A X
1DE7 08             INX
1DE8 7C 0091        INC   NUMBER
1DEB 4D             TST A
1DEC 26 F0          BNE   MOVLFT
1DEE 86 20   SPACE  LDA A #$20
1DF0 BD 0209        JSR   OUTC
1DF3 7D 0091 MOVECS TST   NUMBER
1DF6 27 0A          BEQ   ENDD
1DF8 86 08          LDA A #$08
1DFA BD 0209        JSR   OUTC
1DFD 7A 0091        DEC   NUMBER
1E00 20 F1          BRA   MOVECS
1E02 DE 40   ENDD   LDX   TEMP
1E04 20 CA          BRA   LOOPBA
1E06 DF 40   ADD    STX   TEMP
1E08 7F 0091        CLR   NUMBER
1E0B C6 20          LDA B #$20
1E0D 17      MOVERT TBA
1E0E E6 00          LDA B X
1E10 A7 00          STA A X
1E12 BD 0209        JSR   OUTC
1E15 BD 04BB        JSR   BUFLIM
1E18 7C 0091        INC   NUMBER
1E1B 17             TBA
1E1C 26 F0          BNE   MOVERT+1
1E1E 20 D3          BRA   MOVECS
1E20 7E 045F ERROR  JMP   BERR2
1E23 8D 44   RECALL BSR   ZERCSR
1E25 CE 00BB        LDX   #BUFFER
1E28 BD 073B        JSR   CLASS
1E2B C1 01          CMP B #1
1E2D 26 F1          BNE   ERROR
1E2F BD 0755        JSR   NUMC
1E32 86 3D          LDA A #'=
1E34 A7 00          STA A X
1E36 08             INX
1E37 DF 40          STX   TEMP
1E39 BD 07AB        JSR   LINEC
1E3C 5D             TST B
1E3D 26 E1          BNE   ERROR
1E3F 08             INX
1E40 08             INX
1E41 08             INX
1E42 A6 00   SLOOP  LDA A X
1E44 81 0D          CMP A #$0D
1E46 27 0E          BEQ   ENDR
```

More →

Listing 4 continued.

```
1E48 08              INX
1E49 DF 91           STX   NUMBER
1E4B DE 40           LDX   TEMP
1E4D A7 00           STA A X
1E4F 08              INX
1E50 DF 40           STX   TEMP
1E52 DE 91           LDX   NUMBER
1E54 20 EC           BRA   SLOOP
1E56 DE 40    ENDR   LDX   TEMP
1E58 86 04           LDA A #4
1E5A A7 00           STA A X
1E5C CE 00BB         LDX   #BUFFER
1E5F BD 0485         JSR   PSTR
1E62 DE 40           LDX   TEMP
1E64 6F 00           CLR   X
1E66 7E 1D93  LOOP2  JMP   INPUT
1E69 8C 00BB  ZERCSR CPX   #BUFFER
1E6C 27 08           BEQ   ENDZ
1E6E 86 08           LDA A #8
1E70 BD 0209         JSR   OUTC
1E73 09              DEX
1E74 20 F3           BRA   ZERCSR
1E76 39       ENDZ   RTS
1E77 CE 00BA  ENDRTN LDX   #BUFFER-1
1E7A 5F              CLR B
1E7B 08       LOOP   INX
1E7C 5C              INC B
1E7D A6 00           LDA A X
1E7F 4D              TST A
1E80 26 F9           BNE   LOOP
1E82 D7 8E           STA B CHRCNT
1E84 86 0D           LDA A #$D
1E86 A7 00           STA A X
1E88 7E 03CC         JMP   PLINEC
1E8B 0D              FCB   $0D SET END
   (1E8C)     PRGEND EQU   *

   (0499)            ORG   $499
0499 7E 1D7D  INCHAR JMP   PATCH
049C 81 08    INCHR1 CMP A #8

   (0358)            ORG   $358
0358 CE 1E8C         LDX   #PRGEND

   (16DB)            ORG   $16DB      *** SEE TEXT ***
16DB CE 1E8C         LDX   #PRGEND
                     END
```

Listing 5. Patches required for RTEDIT to work with MIKBUG or SWTBUG.

```
               * THIS PROGRAM PATCHES THE MICRO-WARE EDITOR
               * TO WORK WITH MIKBUG/SWATBUG SYSTEMS
               *
               * MIKBUG EQUATES
E1AC           INEEE    EQU    $E1AC
E1D1           OUTEEE   EQU    $E1D1
E0D0           MON      EQU    $E0D0
               *
0D17                    ORG    $0D17
0D17 E1 D1              FDB    OUTEEE
0D31                    ORG    $0D31
0D31 E1 D1              FDB    OUTEEE
0DCF                    ORG    $0DCF
0DCF E1 D1              FDB    OUTEEE
0D0E                    ORG    $0D0E
0D0E E1 D1              FDB    OUTEEE
               *
0D5D                    ORG    $0D5D
0D5D A0 4A              FDB    CRLF
0DD6                    ORG    $0DD6
0DD6 A0 4A              FDB    CRLF
               *
0D23                    ORG    $0D23
```

More

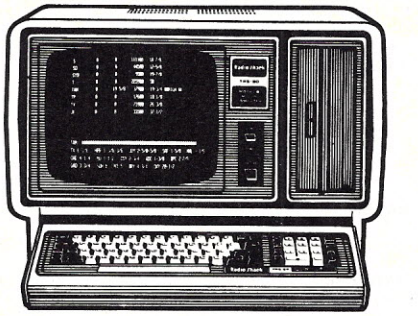
Listing 5 continued.

```
0D23 E1 AC           FDB    INEEE
0CF1                 ORG    $0CF1
0CF1 E1 AC           FDB    INEEE
0CF1                 ORG    $0CF1
0CF1 E1 AC           FDB    INEEE
                *
01A6                 ORG    $01A6
01A6 E0 D0           FDB    MON
                *
A04A                 ORG    $A04A
A04A 86 0A      CRLF LDA A  #$0A
A04C 8D 02           BSR    JOUT
A04E 86 0D           LDA A  #$0D
A050 7E E1 D1   JOUT JMP    OUTEEE
                *
                * THE ABOVE IS ALL THAT IS NECESSARY IF YOU
                * WANT TO USE THE AC-30.
                * THE FOLLOWING WILL ALLOW THE SOURCE TO STAY IN
                * MEMORY WHILE THE COMPILER IS BEING LOADED.
                * THE COMPILER OUTPUT WILL STILL BE TO CASSETTE.
                * NOTE SOME OF THE ABOVE PATCHES WILL NOT BE
                * NECESSARY IF YOU DO THE FOLLOWING. ($0D0E, $0CF1,
                * $0CF9) PATCH COMPILER TO USE MEMORY
                *
                *
                *SECTION TO REWIND MEMORY
                *
0100                 ORG    $0100
0100 7E 20 23        JMP    REWIND
0103            RESTART EQU $0103
                * SECTION TO STORE SOURCE IN MEMORY
                *
0DAB                 ORG    $0DAB
0DAB 86 20           LDA A  #$20
0DAD B7 0D CF        STA A  JMEM+1
0DB0 86 10           LDA A  #$10
0DB2 B7 0D D0        STA A  JMEM+2
0DB5 01              NOP
0DB6 01              NOP
0DB7 01              NOP
0DB8 01              NOP
0DB9 86 02           LDA A  #$02        LOAD DATA START FLAG
0DBB 8D 11           BSR    JMEM
0DBD 8D D8           BSR    PBUFF
0DBF 86 03           LDA A  #$03        LOAD DATA END FLAG
0DC1 8D 0B           BSR    JMEM
0DC3 86 E1           LDA A  #$E1
0DC5 B7 0D CF        STA A  JMEM+1
0DC8 86 D1           LDA A  #$D1
0DCA B7 0D D0        STA A  JMEM+2
0DCD 39              RTS
0DCE 7E E1 D1   JMEM JMP    OUTEEE
0D97            PBUFF EQU   $0D97
                *
                *
                *
                * SECTION TO LOAD NEMORY TO EDITOR
                *
0CEA                 ORG    $0CEA
0CEA 5F              CLR B
0CEB 8D 15      IN1  BSR    JIN
0CED 81 02           CMP A  #$02        START OF DATA
0CEF 26 FA           BNE    IN1
0CF1 8D 0F      IN2  BSR    JIN
0CF3 81 03           CMP A  #$03        END?
0CF5 27 08           BEQ    JEND
0CF7 A7 00           STA A  0,X
0CF9 08              INX
0CFA 5C              INC B
0CFB C1 80           CMP B  #$80        128 BYTES YET?
0CFD 25 F2           BCS    IN2
0CFF 6F 00      JEND CLR    0,X
0D01 39              RTS
0D02 7E 20 00   JIN  JMP    DATAIN
                *
                *
                *
2000                 ORG    $2000
```

More

THE BIGGEST NAME IN LITTLE COMPUTERS™

TRS-80™ Model II—Your Best Buy In a Business Microcomputer



UP TO 15% OFF! on

TRS-80™ computers, software and peripherals

Similar values on all **Radio Shack** merchandise

CALL COLLECT:

915-283-2920

✓214

Van Horn Office Supply

701 W. Broadway -- P O· Box 1060

Van Horn, Texas 79855

MasterCard

Radio Shack®
DEALER

VISA®

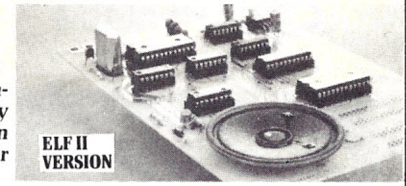
THE NATIONWIDE SUPERMARKET OF SOUND®

NEW!!!-THE ELECTRIC MOUTH*

for S100, Elf II, Apple, TRS-80 Level II*

From $99.95 kit

Now — teach your computer to talk, dramatically increasing the interaction between you and your machine.



That's right: the ELECTRIC MOUTH actually lets your computer talk! Installed and on-line in just minutes, it's ready for spoken-language use in office, business, industrial and commercial applications, in games, special projects, R&D, education, security devices — there's no end to the ELECTRIC MOUTH's usefulness. Look at these features:

* Supplied with 143 words/letters/phonemes/numbers, capable of producing hundreds of words and phrases.
* Expandable on-board up to thousands of words and phrases (just add additional speech ROMs as they become available).
* Four models, which plug directly into S100, Apple, Elf II and TRS-80 Level II computers.
* Get it to talk by using either Basic or machine language (very easy to use, complete instructions with examples included).
* Uses National Semiconductor's "Digitalker" system.
* Includes on-board audio amplifier and speaker, with provisions for external speakers and amplifier.
* Adds a new dimension and excitement to programming; lets you modify existing programs and games to add spoken announcements of results, warnings, etc.
* Installs in just minutes.

"Elf II" and "The Electric Mouth" are reg. trademarks of Netronics R&D Ltd. "Apple" is a reg. trademark of Apple Computer Inc. "TRS-80 Level II" is a reg. trademark of Tandy Corp.

Principle of Operation The ELECTRIC MOUTH stores words in their digital equivalents in ROMs. When words, phrases, and phonemes are desired, they are simply called for by your program and then synthesized into speech. The ELECTRIC MOUTH system requires none of your valuable memory space except for a few addresses if used in memory mapped mode. In most cases, output ports (user selectable) are used.

Spoken Material Included

ONE	SEVENTY	CONTROL	LEFT	POUND
TWO	EIGHTY	DANGER	LESS	PULSES
THREE	NINETY	DEGREE	LESSER	RATE
FOUR	HUNDRED	DOLLAR	LIMIT	RE
FIVE	THOUSAND	DOWN	LOW	READY
SIX	MILLION	EQUAL	LOWER	RIGHT
SEVEN	ZERO	ERROR	MARK	SS
EIGHT	AGAIN	FEET	METER	SECOND
NINE	AMPERE	FLOW	MILE	SET
TEN	AND	FUEL	MILLI	SPACE
ELEVEN	AT	GALLON	MINUS	SPEED
TWELVE	CANCEL	GO	MINUTE	STAR
THIRTEEN	CASE	GRAM	NEAR	START
FOURTEEN	CENT	GREAT	NUMBER	STOP
FIFTEEN	400HERTZ TONE	GREATER	OF	THAN
SIXTEEN	80HERTZ TONE	HAVE	OFF	THE
SEVENTEEN	20MS SILENCE	HIGH	ON	TIME
EIGHTEEN	40MS SILENCE	HIGHER	OUT	TRY
NINETEEN	80MS SILENCE	HOUR	OVER	UP
TWENTY	160MS SILENCE	IN	PARENTHESIS	VOLT
THIRTY	320 MS SILENCE	INCHES	PERCENT	WEIGHT
FORTY	CENTI	IS	PLEASE	
FIFTY	CHECK	IT	PLUS	
SIXTY	COMMA	KILO	POINT	

Also, the complete alphabet, spoken letter by letter.

Continental U.S.A. Credit Card Buyers Outside Conn. TO ORDER CALL TOLL FREE 800-243-7428

To Order From Connecticut Or For Technical Assistance, Etc., Call (203) 354-9375

KB6

NETRONICS R&D LTD. Dept. K5
333 Litchfield Road, New Milford, CT 06776

Please send the items checked below:

- ☐ **S100 "Electric Mouth" kit** **$99.95**
- ☐ **Elf II "Electric Mouth" kit** **$99.95**
- ☐ **Apple "Electric Mouth" kit** **$119.95**
- ☐ **TRS-80 Level II "Electric Mouth" kit** **$119.95**

Add $20.00 for wired & tested units. All plus $3.00 postage & insurance. Conn. res. add sales tax.

Total Enclosed $________

☐ **Personal Check** ☐ **Cashier's Check/Money Order**
☐ **Visa** ☐ **Master Charge (Bank No. ______)**

Acct. No. ________
Signature ________ **Exp. Date** ________
Print Name ________
Address ________
City ________
State ________ **Zip** ________

Hi!

I'M INTO ROBOTICS.
Are you too?

Can't you tell by looking, I'm perfect for computer control, and I can be yours for less than $1500.00! Small price for someone with my "man-rated" features. Of course, if you don't like me (that's RU-2) . . . then B.Y.O.R. Build Your Own Robot. Take my base (now with twice the drive power; that's 2 ft. per sec. with 100 lbs., for only $495.00, Mac), buy my other components and put me together yourself. At the last minute you can decide if I deserve artificial intelligence, or not. Oh, yes. I knew there was something else. Whatever you do, write for Hobby Robotics' *new* catalog. I'm in there some place! I'm really into Robotics. Are you too? Please send $1.00 for prep and handling. (They made me say that.) Yrs. RU-2.

HOBBY ROBOTICS

5070-E Buford Highway • Norcross, Ga. 30071
404/448-0190

✓265

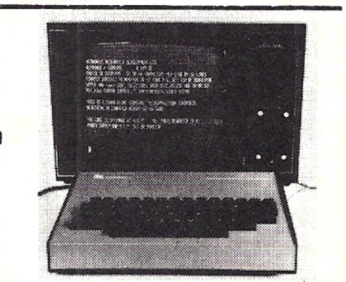
By Netronics

ASCII/BAUDOT, STAND ALONE

Computer Terminal

COMPLETE FOR ONLY . . .

$149^{95}



ASCII/BAUDOT
Computer Terminal Kit

The Netronics ASCII/BAUDOT Computer Terminal Kit is a microprocessor-controlled, stand alone keyboard/terminal requring no computer memory or software. It allows the use of either a 64 or 32 character by 16 line professional display format with selectable baud rate, RS232-C or 20 ma. output, full cursor control and 75 ohm composite video output.

The keyboard follows the standard typewriter configuration and generates the entire 128 character ASCII upper/lower case set with 96 printable characters. Features include onboard regulators, selectable parity, shift lock key, alpha lock jumper, a drive capability of one TTY load, and the ability to mate directly with almost any computer, including the new Explorer/85 and ELF products by Netronics.

The Computer Terminal requires no I/O mapping and includes 1k of memory, character generator, 2 key rollover, processor controlled cursor control, parallel ASCII/BAUDOT to serial conversion and serial to video processing—fully crystal controlled for superb accuracy. PC boards are the highest quality glass epoxy for the ultimate in reliability and long life.

VIDEO DISPLAY SPECIFICATIONS

The heart of the Netronics Computer Terminal is the microprocessor-controlled Netronics Video Display Board (VID) which allows the terminal to utilize either a parallel ASCII or BAUDOT signal source. The VID converts the parallel data to serial data which is then formatted to either RS232-C or 20 ma. current loop output, which can be connected to the serial I/O on your computer or other interface, i.e., Modem.

When connected to a computer, the computer must echo the character received. This data is received by the VID which processes the information, converting to data to video suitable to be displayed on a TV set (using an RF modulator) or on a video monitor. The VID generates the cursor, horizontal and vertical sync pulses and performs the housekeeping relative to which character and where it is to be displayed on the screen.

Video Output: *1.5 P/P into 75 ohm (EIA RS-170)* • **Baud Rate:** *110 and 300 ASCII* • **Outputs:** *RS232-C or 20 ma. current loop* • **ASCII Character Set:** *128 printable characters—*

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

BAUDOT Character Set: *A B C D E F G H I J K L M N O P Q R S T U V W X Y Z - ? : • 3 $ # ( ) . , 9 0 1 4 ! 5 7 ; 2 / 6 8* • **Cursor Modes:** *Home, Backspace, Horizontal Tab, Line Feed, Vertical Tab, Carriage Return. Two special cursor sequences are provided for absolute and relative X-Y cursor addressing* • **Cursor Control:** *Erase, End of Line, Erase of Screen, Form Feed, Delete* • **Monitor Operation:** *50 or 60Hz (jumper selectable.*

Continental U.S.A. Credit Card Buyers Outside Connecticut

CALL TOLL FREE 800-243-7428

To Order From Connecticut Or For Technical Assistance, Etc. **Call (203) 354-9375**

KB6

Netronics R&D Ltd., Dept.
333 Litchfield Road, New Milford, CT 06776

Please send the items checked below—

- ☐ **Netronics Stand Alone ASCII Keyboard/Computer Terminal Kit, $149.95** plus $3.00 postage & handling.
- ☐ **Deluxe Steel Cabinet** for Netronics Keyboard/Terminal In Blue/Black Finish, **$19.95** plus $2.50 postage and handling.
- ☐ **Video Display Board Kit** alone (less keyboard), **$89.95** plus $3 postage & handling.
- ☐ **12" Video Monitor** (10 MHz bandwidth) fully assembled and tested, **$139.95** plus $5 postage and handling.
- ☐ **RF Modulator Kit** (to use your TV set for a monitor), **$8.95** postpaid.
- ☐ **5 amp Power Supply Kit** In Deluxe Steel Cabinet (±8VDC @ 5 amps, plus 6-8 VAC), **$39.95** plus $2 postage & handling.

Total Enclosed (Conn. res. add sales tax) $________
By—
☐ Personal Check ☐ Cashiers Check/Money Order
☐ **Visa** ☐ **Master Charge** (Bank # _ _ _ _)

Acct. # ________
Signature ________ Exp. Date ________
Print Name ________
Address ________
City ________
State ________ Zip ________

☐ **Send Me More Information**

✓See List of Advertisers on page 210

Listing 5 continued.

```
2000 FF 20 0D   DATAIN  STX     XSAV+1
2003 CE 20 3B   LDATA1  LDX     #DATA
2006 A6 00              LDA A   0,X
2008 08                 INX
2009 FF 20 04           STX     LDATA1+1
200C CE 00 00   XSAV    LDX     #0000
200F 39                 RTS
                *
                *
                *
2010 FF 20 20   DATAOUT STX     XSAV2+1
2013 CE 20 3B   LDATA2  LDX     #DATA
2016 A7 00              STA A   0,X
2018 08                 INX
2019 FF 20 14           STX     LDATA2+1
201C BD E1 D1           JSR     OUTEEE
201F CE 00 00   XSAV2   LDX     #0000
2022 39                 RTS
                *
                *
                *
2023 4F         REWIND  CLR A
2024 C6 32              LDA B   #$32
2026 CE 20 3B           LDX     #DATA
2029 FF 20 04           STX     LDATA1+1
202C FF 20 14           STX     LDATA2+1
202F 7E 01 03           JMP     RESTART
                *
                *
2032                    RMB     $9         ALTERNATE CRLF ROUTINE AREA
                *
203B            DATA    EQU     *
                        END
```

Listing 6. Patches required for A/BASIC to work with MIKBUG or SWTBUG.

```
                * THIS PROGRAM PATCHES THE MICRO-WARE COMPILER
                * TO WORK WITH MIKBUG/SWATBUG SYSTEMS
                *
                * MIKBUG EQUATES
E1AC            INEEE   EQU     $E1AC
E1D1            OUTEEE  EQU     $E1D1
E0D0            MON     EQU     $E0D0
E0BF            OUT2H   EQU     $E0BF
                *
16E5                    ORG     $16E5
16E5 E1 AC              FDB     INEEE
18E5                    ORG     $18E5
18E5 E1 AC              FDB     INEEE
18ED                    ORG     $18ED
18ED E1 AC              FDB     INEEE
1917                    ORG     $1917
1917 E1 AC              FDB     INEEE
                *
16EE                    ORG     $16EE
16EE A0 4A              FDB     CRLF
17D0                    ORG     $17D0
17D0 A0 4A              FDB     CRLF
1951                    ORG     $1951
1951 A0 4A              FDB     CRLF
                *
16E2                    ORG     $16E2
16E2 E1 D1              FDB     OUTEEE
17C9                    ORG     $17C9
17C9 E1 D1              FDB     OUTEEE
1902                    ORG     $1902
1902 E1 D1              FDB     OUTEEE
190B                    ORG     $190B
190B E1 D1              FDB     OUTEEE
1925                    ORG     $1925
1925 E1 D1              FDB     OUTEEE
                *
0827                    ORG     $0827
```

More

Listing 6 continued.

```
0827 E0 D0              FDB     MON
                  *
A04A                    ORG     $A04A
A04A 86 0A        CRLF  LDA A   #$0A
A04C 8D 02              BSR     JOUT
A04E 86 0D              LDA A   #$0D
A050 7E E1 D1     JOUT  JMP     OUTEEE
                  *
                  * THIS WAS A TOUGH ONE
                  *
16D1                    ORG     $16D1
16D1 BD A0 4A           JSR     CRLF
16D4 96 42              LDA A   $42
16D6 26 2C              BNE     CONT
16D8 39                 RTS
16D9 EB 00        PATCH ADD B   0,X
16DB 7E E0 BF           JMP     OUT2H
16DE 7E 16 D9           JMP     PATCH
1704              CONT  EQU     $1704
                  *
                  * THIS WAS EVEN TOUGHER
                  * FIXES OPT S
                  *
0F11                    ORG     $0F11
0F11 01                 NOP
0F12 01                 NOP
                  *
                  *
                  * THE ABOVE IS ALL THAT IS NECESSARY IF YOU
                  * WANT TO USE THE AC-30
                  * THE FOLLOWING WILL ALLOW THE SOURCE TO STAY IN
                  * MEMORY WHILE THE COMPILER IS BEING LOADED
                  * THE COMPILER OUTPUT WILL STILL BE TO CASSETTE.
                  * NOTE SOME OF THE ABOVE PATCHES WILL NOT BE
                  * NECESSARY IF YOU DO THE FOLLOWING. ($18E5, $18ED)
                  * THE EDITOR WILL HAVE TO BE PATCHED TO MATCH
                  *
                  *SECTION TO REWIND MEMORY
                  *
00FB                    ORG     $00FB
00FB BD 20 00     BACK  JSR     REWIND
00FE 20 10              BRA     FWD
0100 20 F9              BRA     BACK
0110              FWD   EQU     $0110
                  *
                  *
                  *
                  * SECTION TO LOAD MEMORY TO COMPILER
                  *
18DE                    ORG     $18DE
18DE 5F                 CLR B
18DF 8D 15        IN1   BSR     JIN
18E1 81 02              CMP A   #$02        START OF DATA
18E3 26 FA              BNE     IN1
18E5 8D 0F        IN2   BSR     JIN
18E7 81 03              CMP A   #$03        END?
18E9 27 08              BEQ     JEND
18EB A7 00              STA A   0,X
18ED 08                 INX
18EE 5C                 INC B
18EF C1 80              CMP B   #$80        128 BYTES YET?
18F1 25 F2              BCS     IN2
18F3 6F 00        JEND  CLR     0,X
18F5 39                 RTS
18F6 7E 20 00     JIN   JMP     DATAIN
                  *
                  *
                  *
2000                    ORG     $2000
2000 FF 20 16     DATAIN STX    XSAV+1
2003 CE 20 3B     LOAD  LDX     #DATA
2006 A6 00              LDA A   0,X
2008 08                 INX
2009 81 1A              CMP A   #$1A        END?
200B 26 05              BNE     STORE
200D CE 20 3B     REWIND LDX    #DATA
2010 86 03              LDA A   #$03
2012 FF 20 04     STORE STX     LOAD+1
```

More

✓See List of Advertisers on page 210

Listing 6 continued.

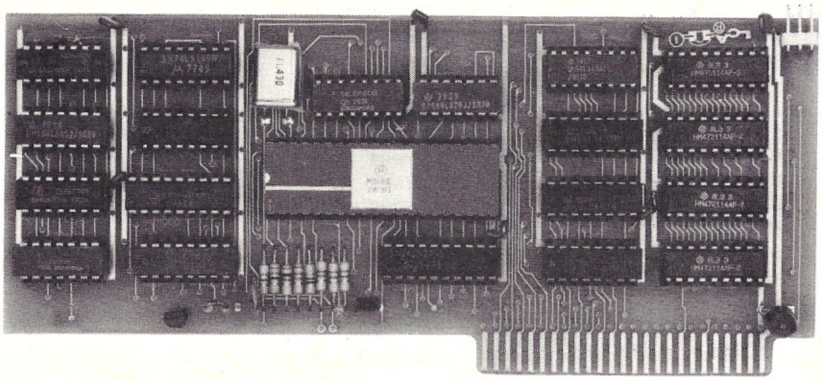
```
2015 CE 00 00  XSAV    LDX     #0000
2018 39                RTS
203B           DATA    EQU     $203B
               *
               * IT WILL BE NECESSARY TO KEEP THE CRLF IN
               * $A04A ANYTIME YOU ARE RUNNING PROGRAMS THAT
               * HAVE BEEN COMPILED BY THE A/BASIC COMPILER
               * A BETTER METHOD IS TO INSERT THE CRLF ROUTINE
               * AT THE END OF THE COMPILED CODE.  THE COMPILER
               * TELLS YOU WHERE THE END IS.  THEN SEARCH OUT
               * THE JUMPS TO $A04A AND PATCH TO THE NEW CRLF
               * ROUTINE.  (FOR AN EXCELLENT SEARCH ROUTINE SEE
               * MAR 1978 73 MAGAZINE)
                       END
```

Listing 7. Program to use miniFlex text files.

```
               *
               * THIS PROGRAM PATCHES THE MICRO-WARE COMPILER
               * TO WORK WITH MIKBUG/SWATBUG SYSTEMS
               * AND TO ALLOW THE USE OF FLEX TEXT FILES.
               * THE TSC EDITOR CAN GENERATE THE FILES.
               * OUTPUT WILL BE TO MEMORY STARTING AT $2000.
               * PROGRAMS WITH ORG=$2000 CAN BE RUN THERE, ELSE
               * RELOCATE THEM TO ACTUAL ORG ADDRESS.
               * MAKE THE FOLLOWING CHANGES TO THE
               * PROGRAM TO THE ABASIC ORIGINAL (CASETTE VERSION).
               * MAKE ABASIC A COMMAND THEN USE ABASIC,<TEXT FILE>
               * THE CRLF IS ROUTED TO FLEX.  TO ALLOW THE
               * COMPILED PROGRAMS TO RUN INDEPENDANT OF FLEX
               * IT WILL BE NECESSARY TO SEARCH OUT THE JUMPS
               * TO PCRLF AND PUT A CRLF ROUTINE AT THE END OF
               * THE COMPILED PROGRAM.
               *
               *
               *
               * MIKBUG EQUATES
E0D0           MON     EQU     $E0D0
E1AC           INEEE   EQU     $E1AC
E1D1           OUTEEE  EQU     $E1D1
E07E           PDATA1  EQU     $E07E
               *
               * FIX MON JUMP FOR COMPILED PROGRAMS
               *
0A1A                   ORG     $0A1A
0A1A 86 E0             LDA A   #$E0
0A1C C6 D0             LDA B   #$D0

               *
               * FLEX EQUATES
               *
7112           PUTCHR  EQU     $7112
710F           GETCHR  EQU     $710F
7118           PSTRNG  EQU     $7118
7139           OUTHEX  EQU     $7139
711E           PCRLF   EQU     $711E
7103           WARMS   EQU     $7103
7127           GETFIL  EQU     $7127
712D           SETEXT  EQU     $712D
713C           RPTERR  EQU     $713C
7806           FMS     EQU     $7806
7803           FMSCLS  EQU     $7803
7740           FCB     EQU     $7740
               *
0100           ABASIC  EQU     $0100
               *

1917                   ORG     $1917
1917 E1 AC             FDB     INEEE
               *
0103                   ORG     $0103
0103 76 1F             FDB     REWIND
```

More →

NOW THAT YOU'RE SEEING DOUBLE YOU NEED MORE ROOM...



DOUBLEVISION™
80 x 24 Video Display from Computer Stop

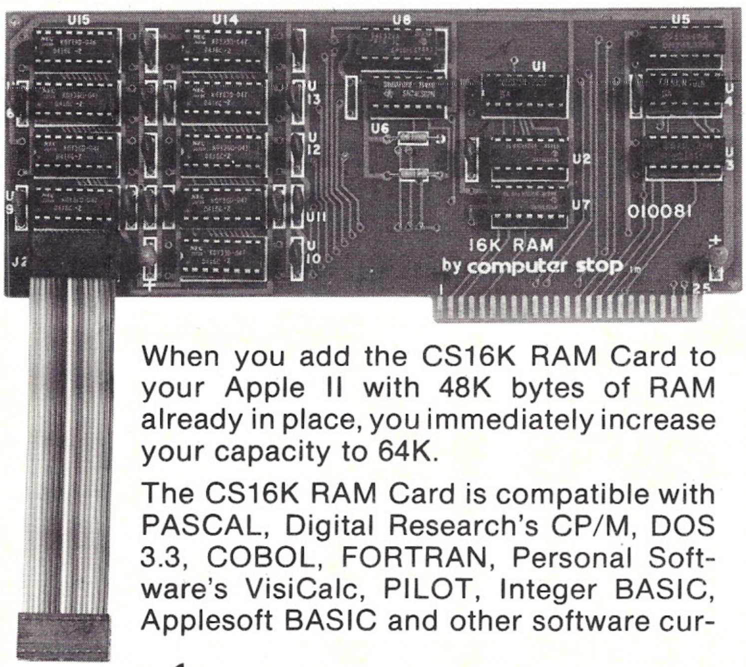
DoubleVision is the time-tested upper-and-lower-case Video Display. And now it has a host of new features. Like the availability of a completely commented source listing of software and hardware schematics. Or the inclusion of a Sysgen program to select type of cursor, monitor band width and display of control characters.

Moreover, DoubleVision is compatible with CCA Data Management (and full screen mapping), Easywriter Professional System, Apple Pie 2.0, PASCAL, Z-80 Softcard, MAGIC-WAND and WORDSTAR.

It's the only 80-column card that has software on disk making it easy to provide updates. In fact, updates are automatically mailed to our customers and *billed* for cost of disk and shipping only. In short, DoubleVision is adaptable to meet your needs.

The price? It's just $295.

...AND HERE'S MORE ROOM!



The CS16K RAM Card
(compatible with PASCAL, CP/M®, DOS 3.3 and other Apple software)

When you add the CS16K RAM Card to your Apple II with 48K bytes of RAM already in place, you immediately increase your capacity to 64K.

The CS16K RAM Card is compatible with PASCAL, Digital Research's CP/M, DOS 3.3, COBOL, FORTRAN, Personal Software's VisiCalc, PILOT, Integer BASIC, Applesoft BASIC and other software currently used with APPLE II.

What's more, the CS16K RAM Card is backed by the Computer Stop reputation for excellent service. In fact, **we've doubled the warranty to 180 days,** instead of the usual 90.

And we've saved the best 'til last. The Price. It's only $195.

computer stop Your best stop for Apple products.

16919 Hawthorne Blvd.
Lawndale, CA 90260
(213) 371-8500
Available at select Apple dealers.

283

Apple is a trademark of Apple Computer, Inc.
CP/M is a trademark of Digital Research

2545 W. 237th Street
Torrance, CA 90505
(213) 539-7670
Dealers: Ask about special-priced demo unit.

See List of Advertisers on page 210

Listing 7 continued.

```
                         *
17D0                            ORG     $17D0
17D0 71 1E                      FDB     PCRLF

1951                            ORG     $1951
1951 71 1E                      FDB     PCRLF
                         *
16E1                            ORG     $16E1
16E1 7E 71 12                   JMP     PUTCHR
16E4 7E 71 0F                   JMP     GETCHR
16E7 7E 17 0F                   JMP     OUT4HS
16EA 7E 17 13                   JMP     OUT2HS
16ED 7E 71 1E                   JMP     PCRLF
16F0 7E 71 18                   JMP     PSTRNG
                         *
170F                            ORG     $170F
170F BD 71 39            OUT4HS JSR     OUTHEX
1712 08                         INX
1713 BD 71 39            OUT2HS JSR     OUTHEX
1716 20 42                      BRA     OUTSP
175A                            ORG     $175A
175A 86 20               OUTSP  LDA A   #$20
175C 08                         INX
175D 7E 71 12                   JMP     PUTCHR


                         *
17C9                            ORG     $17C9
17C9 E1 D1                      FDB     OUTEEE
1902                            ORG     $1902
1902 E1 D1                      FDB     OUTEEE
190B                            ORG     $190B
190B E1 D1                      FDB     OUTEEE
1925                            ORG     $1925
1925 E1 D1                      FDB     OUTEEE
                         *
0827                            ORG     $0827
0827 E0 D0                      FDB     MON
                         *
                         * GO CLOSE ALL FLEX FILES
                         *

1689                            ORG     $1689
1689 BD 76 2C                   JSR     CLOSE
168C 01                         NOP
168D 01                         NOP
168E 01                         NOP

                         *
                         * CARRIAGE RETURN LINE FEED ROUTINE
                         *
                         * LOAD OUTPUT TO MEMORY STARTING AT $2000
                         *

16A7                            ORG     $16A7
16A7 CE 00 90                   LDX     #$0090     GET BUFFER ADDRESS
16AA E6 00                      LDA B   0,X        GET LENGTH
16AC 08                         INX                SKIP TO DATA
16AD 08                         INX
16AE 08                         INX
16AF A6 00               OUT1   LDA A   0,X        GET DATA BYTE
16B1 FF 16 C3                   STX     OUT3+1     STORE POINTER
16B4 CE 20 00            OUT2   LDX     #$2000     GET MEMORY ADDRESS
16B7 A7 00                      STA A   0,X        STORE DATA
16B9 08                         INX
16BA 8C 6F FF                   CPX     #$6FFF     ANY MEMORY LEFT?
16BD 27 0B                      BEQ     OUT4       NO-REPORT MEMORY FULL
16BF FF 16 B5                   STX     OUT2+1     YES-STORE MEMORY POINTER
16C2 CE 00 00            OUT3   LDX     #$000      RESTORE BUFFER POINTER
16C5 08                         INX
16C6 5A                         DEC B
16C7 26 E6                      BNE     OUT1       ALL DONE, NO-GOBACK
16C9 39                         RTS
16CA CE 16 D6            OUT4   LDX     #MSG1
16CD BD E0 7E                   JSR     PDATA1
16D0 BD 76 2C                   JSR     CLOSE
16D3 7E 71 03                   JMP     WARMS
16D6 54                  MSG1   FCC     /TOO BIG/
16DD 04                         FCB     4
```

More →

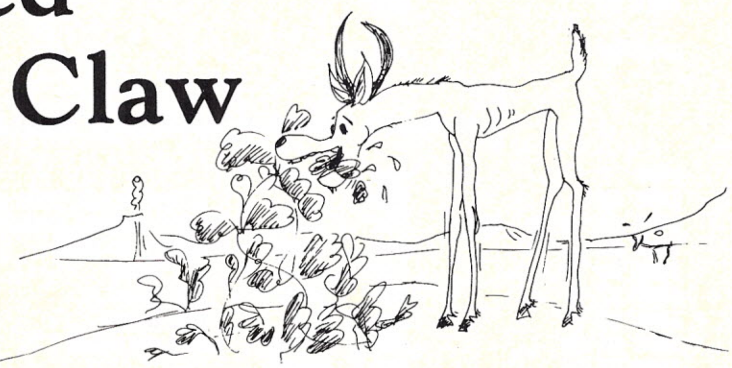
Listing 7 continued.

```
                    *
                    * THIS WAS TOUGH
                    * FIXES OPT S
                    *
0F11                        ORG     $0F11
0F11 01                     NOP

0F12 01                     NOP
                    *
                    *
                    * SECTION TO LOAD FLEX TEXT FILE TO COMPILER
                    *
18DE                        ORG     $18DE
18DE 5F                     CLR B           CLEAR B FOR BYTE COUNT
18DF FF 18 F6       IN2     STX     XSAV+1  SAVE BUFFER PONTER
18E2 CE 77 40               LDX     #FCB    POINT TO FCB
18E5 BD 78 06               JSR     FMS     GET DATA
18E8 27 0B                  BEQ     XSAV    NO ERROR
18EA A6 01                  LDA A   1,X     GET ERROR
18EC 81 08                  CMP A   #$08    END OF FILE?
18EE 27 03                  BEQ     JRET2
18F0 7E 76 39               JMP     ERROR

18F3 86 03          JRET2   LDA A   #$3
18F5 CE 00 00       XSAV    LDX     #0000   RESTORE X
18F8 A7 00                  STA A   0,X
18FA 81 03                  CMP A   #$03    END?
18FC 27 06                  BEQ     JEND
18FE 08                     INX
18FF 5C                     INC B
1900 C1 80                  CMP B   #$80    128 BYTES YET?
1902 25 DB                  BCS     IN2
1904 6F 00          JEND    CLR     0,X
1906 39                     RTS
                    *
                    * OPEN TEXT FILE FOR ABASIC

                    *
                    * REWIND FILE
                    *
761F 86 05          REWIND  LDA A   #$5     REWIND CODE
7621 CE 77 40               LDX     #FCB
7624 A7 00                  STA A   0,X
7626 BD 78 06               JSR     FMS
7629 26 0E                  BNE     ERROR
762B 39                     RTS
                    *
                    * ERROR SECTION
                    *

7600                        ORG     $7600
7600 CE 77 40       START   LDX     #FCB    POINT TO FCB
7603 BD 71 27               JSR     GETFIL  GET FILE SPEC
7606 25 31                  BCS     ERROR   ANY ERRORS
7608 CE 77 40               LDX     #FCB
760B 86 01                  LDA A   #1      SET DEFAULT EXT.
760D BD 71 2D               JSR     SETEXT
7610 CE 77 40               LDX     #FCB
7613 86 01                  LDA A   #1      OPEN FOR READ
7615 A7 00                  STA A   0,X
7617 BD 78 06               JSR     FMS
761A 26 1D                  BNE     ERROR
761C 7E 01 00               JMP     ABASIC

762C 86 04          CLOSE   LDA A   #4      CLOSE FILE CODE
762E CE 77 40               LDX     #FCB
7631 A7 00                  STA A   0,X
7633 BD 78 06               JSR     FMS
7636 26 01                  BNE     ERROR
7638 39                     RTS
                    *
                    *
7639 BD 71 3C       ERROR   JSR     RPTERR
763C BD 78 03               JSR     FMSCLS  CLOSE FILES
763F 7E 71 03               JMP     WARMS   GOTO FLEX
                            END     START
```


✓See List of Advertisers on page 210

You'll watch a struggle for survival unfold in this simulation program for the OSI C2-4P.

Nature Red In Tooth and Claw



By Edward H. Carlson

With a snarl of hunger, the saber-toothed tiger leaped toward the....

Don't get too emotionally involved in this little drama of nature—after all, it's just a simulation. More about the sabertooths later. This article describes a two-level ecological niche in which a population of herbivores (browsers) feeds on foliage which grows back at a rate that depends on environmental factors.

The game resembles Conway's Life in several respects. It does its work out in the open, on a screen display that amuses the eye and prods the imagination. The two-level ecological system provides plenty of variety, from ecology in delicate balance to spectacular blooms of growth followed by sudden crashes from which there may, or may not, be recovery.

There are many adjustable parameters in the system, including rates of growth and nourishment, morphology of the leaves and mobility of the browsers. You can even add another trophic level if you wish: sabertooths pouncing on the browsers. In fact, so rich is the modeling of the system that even seemingly insignificant changes in the program can produce startling changes in the stability or rate of growth of the populations.

Any program that must continuously refresh a complicated screen display requires sophisticated programming if it is to run at a reasonable speed. I have found and will discuss some key techniques that reduce running time in BASIC programs to a minimum. At least they work for my machine's version of BASIC, and probably for most BASICs.

I have an Ohio Scientific C2-4P with 16K of programmable memory. It uses OSI Microsoft BASIC-in-ROM version 1.0, rev. 3.2. The screen has 2048 display cells, arranged in 32 rows of 64 characters each. The Browser Ecology program can easily be adapted to other screen sizes and to smaller-memory machines.

Browser's World is a Torus

The program listing is the simulation of the ecological system. The screen topology is the torus, or doughnut, often used in screen games; this is accomplished in lines 9, 42 and 83, which shift objects going off the bottom of the screen to the corresponding point at the top.

Due to the continuous nature of the screen memory array, objects shifted off the right edge of the screen appear at the left edge one line lower. Thus, the display is actually a spiral of 32 turns wound around the torus, much as a strip of tape could be spirally wound around an innertube, closing to a match at the ends. In such an arrangement, the cells at the four corners of the screen are actually adjacent to each other.

The Simulation

The simulation begins by randomly putting a number of browsers on the screen, each represented by the letter B, and a number of leaf cells, each represented by another character. (An X or an O would be appropriate, but I used character 187 from those available on the OSI graphics chip.) From that point on, browser-move-and-eat seasons alternate with leaf-growing seasons.

Information about browser J is stored in the element P of the array P(J). The integer part of P is the browser's position, and the decimal fraction part of P is the status of his energy reserve, E. The position of leaf cell J is stored in another array, PL(J).

The screen memory is an integral part of the program, not just a display array. Before a browser can be moved to a given cell on the screen, the cell is checked by a PEEK (line 10) to see if it is already occupied by a browser. If not, he is moved, and if the cell is part of a leaf, he is fed a unit F of nourishment (line 20). Likewise, a growing leaf can only expand into an empty cell on the screen.

In the browser-moving season, the list of browsers is scanned. As each entry is inspected, the browser is removed if dead. Otherwise, he is moved, then fed if he lands on a leaf and reproduced if his energy is over the reproduction threshold.

The list of browsers is kept compact by "garbage collection." Dead browsers in the middle of the list are replaced by live ones from the end. Newly born browsers are added at the end of the list. In each case, the number of browsers (NB) is updated. A flag (P = −1) is set at the end of the browser list, and, for speed in running, only that portion of the array P(J) containing live browsers is scanned each cycle.

During the leaf-growing season, the leaf list PL(J) is processed. I am treating these in a less dynamic way than the browsers. Growth is limited

Edward H. Carlson is a physics professor at Michigan State University, East Lansing, MI 48824.

to two possible directions and is independent of any individual cell food reserves. However, the growth rate is sensitive to overall crowding of the leaves (line 71).

Other schemes to treat leaves are possible, and you can easily modify the program to treat the leaves in a manner analogous to the browsers.

The dynamics of the simulation is largely determined by the browser energy E. As it is fed, its stored energy value rises toward a reproduction threshold, G (growth), at which it splits into two browsers. Each browser's energy is decreased by a factor Z at each cycle. If not augmented by feeding, the energy of a browser will decrease below the fatal threshold, D, after a number of seasons, and the browser dies.

Fast BASIC

I recently ran a number of timing tests on my machine and was surprised by the results. Several techniques for speeding up BASIC emerged. First, any number in a BASIC statement that is in digital form (rather than as a variable name) must be converted to floating point form every time it is encountered by the interpreter. This is a time-consuming operation, so variable names should be used instead of explicit constants.

Even the line number addresses in GOTO and GOSUB expressions must be decoded each time they are encountered, so there is an advantage to making them as short as possible. (This BASIC does not allow variables here.)

The line numbers at the beginning of each line are exempt from this discussion, as they are decoded once-for-all when the program is entered into source memory.

When GOTO or GOSUB is encountered, the interpreter must search for the line described, starting at the beginning of the program. The second surprise is that this requires a rather long time, 0.85 ms for each line examined. This time mounts up very rapidly, especially if the GOTO is called many times from an inner loop. Naturally, the closer the sought-for line is to the top of the program, the faster it is found.

These discoveries have changed my style of writing BASIC programs completely. I used to write a BASIC program in the same order as a FORTRAN program, with the introduction, initialization, explanation of variables and so on at the front, followed by the main structure and with the subroutines collected at the end. Now, for BASIC programs, I reverse this completely, putting initialization at the end, main loops in the middle and subroutines and GOTO target lines first.

The style of the program after line 1000, where initialization starts, is quite different from the lines before 1000, where the main loop and subroutines are. Every effort is made to reduce the number of lines by putting multiple commands on a single line, using the colon separator. Sections whose lines are called from inner loops are put first, followed by those called less often.

Other, less important, time-saving techniques are also used. In an IF statement, make the branch most often taken the false one, if possible. In such cases, the part of the IF statement following THEN is not interpreted at all.

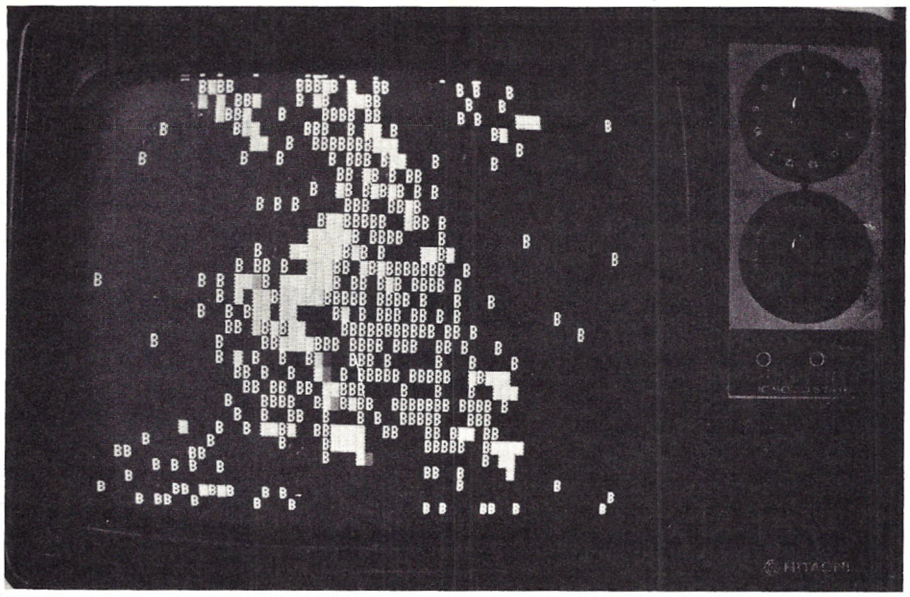
Variable names longer than two characters are redundant in this version of BASIC, so when variables are first introduced, they are presented as full descriptive words, such as SPACE, but truncated to the first two letters, SP, when used in time-sensitive parts of the code. Exceptions may have to be made. For example, LEAF, when truncated in an IF... THEN statement becomes ...LE THEN..., which is illegal, being interpreted as ...LET HEN....

A central feature of this program is the randomness of the browser movement and leaf growth. However, calling the random function, RND, is needlessly expensive (4.5 ms for each toss of the coin), so RND is called infrequently. Randomness is achieved mostly by using random initial conditions. Continuous shuffling of the lists P(J) and PL(J) by the garbage collection process maintains randomness as the program progresses.

There are a few tricks that are *not* worth using for speed-up purposes; these include leaving all spaces out of the lines and making all variables single letters. After all, clarity in reading the program is worth something, and the time saved would be small anyway.

Results and Variations

Now let's get down to cases. What



Browsers are represented by B, leaves by the filled-in areas. A cloud of browsers away from all leaves is a remnant population starving after having eaten all the food in their vicinity. The TV monitor is a Hitachi modified by a Pickles and Trout board.

Browser Ecology simulation program. Browsers feed on leaves. To minimize running time an inverted progamming scheme is used: Subroutines are placed at the top, followed by the main loop, with initialization routines last.

```
1 REM ***** BROWSER ECOLOGY *****
2 REM
3 REM REMOVE ALL SELF STANDING REM LINES UP TO LINE 1000
4 REM
7 GOTO 1000
```

More

happens when you run this simulation, and what can you find out by making various changes? When the program is run as is, it takes 18 seconds (on my machine) to complete the first "year" comprising a browser-moving season for 50 browsers and a growing season for 200 leaves.

After quite a few such years, a stable cyclic history is established. Picking up the cycle when the screen is nearly empty, you can see leaves start growing rapidly, diagonally down the screen, perhaps curling around the torus more than once. When the screen is almost full, the browser population explodes, eating greedily until the screen is nearly full of browsers and empty of leaves.

Most of the browsers then starve to death until the screen is nearly empty again, ready for the leaves to begin their period of rapid growth. The simulation can continue in this cyclic manner for many hours (real time).

Most of the changes you can make in the program fall into two broad classes—those affecting browser mobility and those affecting leaf shape (morphology). Although food ratio F, reproduction threshold G, death threshold D and metabolism Z are all separate and interesting parameters,

Listing continued.

```
8 REM MOVE BROWSER
9 IF S<Q OR S>Q+SC THEN S=S+SC:IF S>Q+SC THEN S=S-2*SC
10 LK=PEEK(S):IF LK=B THEN S=S+D(ND):GOTO9
19 REM BROWSER FEEDS
20 IF LK=LEAF THEN E=E+F
23 REM BROWSER STARVES TO DEATH
24 IF E<D THEN P=P(NB):P(NB)=-1:NB=NB-1:R=INT(P):E=P-R:S=Q+R:RETURN
28 REM BROWSER SURVIVES
29 IF E<G THEN RETURN
30 REM BROWSER REPRODUCES
40 NB=NB+1:SN=S+D(ND)
42 IF SN<Q OR SN>Q+SC THEN SN=SN+SC:IF SN>Q+SC THEN SN=SN-2*SC
44 IF PEEK(SN)=B THEN SN=SN+D(ND):GOTO42
47 P(NB)=SN-Q+ES:E=ES:P(NB+1)=-1:RETURN
69 REM GROWING SEASON FOR THE LEAVES
70 J=INT(RND(8)*10)+1:JJ=NL:IF NL>.9*MAXL THEN GOSUB 90:RETURN
71 DJ=INT(2*MAXL/(MAXL-NL))
73 IF J>=JJ THEN NL=JJ:RETURN
77 PL=PL(J):IFPEEK(Q+PL)<>LEAFTHENPL(J)=PL(JJ):JJ=JJ-1:J=J+DJ:GOTO77
83 PL=PL+1:IF PL>SC THEN PL=PL-SC
86 IFPEEK(Q+PL)<>SPTHENPL=PL+1:IFPEEK(Q+PL)<>SPTHENPL=PL+63:GOTO83
88 PL(JJ+1)=PL:JJ=JJ+1:POKE Q+PL,LEAF:J=J+DJ:GOTO 73
89 REM GARBAGE COLLECTION FOR LEAF ARRAY
90 JL=NL:FOR LL=1 TO NL
94 IF PEEK(PL(LL))<>LEAF THEN PL(LL)=PL(JL):JL=JL-1
96 IF LL>=JL THEN LL=NL+1:NL=JL
98 NEXT:RETURN
200 REM ** MAIN LOOP **
202 FOR I=1 TO NL
210 P=P(I):IF P=-1 THEN 300
219 REM T IS THE OLD ADDRESS OF THE BROWSER, S THE NEW
220 R=INT(P):T=Q+R:E=(P-R)*Z:S=T+D(ND):ND=ND+1
225 IF ND>8 THEN ND=1
230 GOSUB 9:POKE T,SP:POKE S,B:P(I)=S-Q+E
240 NEXT
300 ND=INT(RND(8)*8)+1:GOSUB 70
999 GOTO 202
1000 REM ***** BROWSER ECOLOGY *****
1002 REM
1004 REM       EDWARD H. CARLSON
```

More

Listing continued.

```
1006 REM       3872 RALEIGH DR.
1008 REM       OKEMOS MICHIGAN 48864
1010 REM
1011 Q=53248:SC=2047:REM Q IS UPPER LH CORNER OF SCREEN
1012 B=66:LEAF=187:SPACE=32:REM GRAPHICS CHARACTERS
1015 ES=.45:REM ENERGY AT THE START
1016 D=.3:REM DEATH THRESHOLD
1017 G=.5:REM REPRODUCTION THRESHOLD
1019 Z=.9:REM METABOLISM RATE
1021 F=.4:REM FOOD INCREMENT
1023 MAXB=700:REM NOTE: BOTH MAXB AND MAXL ARE STORED AS "MA"
1025 NB=50:REM NUMBER OF BROWSERS
1027 DIM P(MAXB):REM ARRAY HOLDS BROWSER POSITION AND ENERGY
1029 REM SET UP THE BROWSERS ON THE SCREEN
1030 FOR I=1 TO NB
1032 P=INT(RND(8)*(SC-128))+64
1034 IF PEEK(Q+P)<>SP THEN P=P+1:GOTO 1034
1035 IF P>2047 THEN P=P-64:GOTO 1034
1036 POKE Q+P.B
1037 P(I)=P+ES
1039 NEXT:P(NB+1)=-1:REM FLAG FOR END OF BROWSER LIST
1040 MAXL=1200
1041 NL=200:REM NUMBER OF LEAVES
1042 DIM PL(MAXL)
1050 REM SET UP THE LEAVES
1052 FOR I=1 TO NL
1054 P=INT(RND(8)*(SC-128))+64
1056 IF PEEK(Q+P)<>SP THEN P=P+1:GOTO 1056
1058 POKE Q+P,LE
1060 PL(I)=P
1070 NEXT
1099 REM:DIRECTIONS BROWSERS CAN MOVE
1100 DATA 1.-1.64.-64,63.-63.65.-65
1110 FOR I=1 TO 8
1120 READ D(I):NEXT
1121 D(0)=1
1990 ND=1
1994 REM: DJ USED TO SLOW LEAF GROWTH AT LARGE POPULATION
1995 DJ=10
2000 GOTO202
```

collectively they determine how quickly the browsers can diffuse in a food-rich area, how explosively the population will grow and how rapidly it will crash after the food is gone.

Experimenting with these parameters is rewarding. In fact, it is easy to adjust them into a range where there is no recovery after the initial crash, or where there is exponential growth until the array size is exceeded.

The parameters F, G, D and ES are not completely independent; only the ratio of each of them to one given parameter, say F, is significant. (Be careful that G does not exceed 1, because E is taken to be the decimal fractional part of P.)

On the other hand, one can play with the initial numbers of browsers and leaf cells to search for stable populations in which oscillations do not occur.

Leaf morphology is important. Changing 63 to 62 in line 86 will yield vertical leaves. These act much like the diagonal leaves discussed above, but have a practical disadvantage if they grow into a ring around the torus. Then the growing season for leaves may become very long in real time because of a nearly infinite loop in lines 83 to 86.

✓See List of Advertisers on page 210

Exploring other possible morphologies will yield different degrees of stability. The ability of roaming browsers to find newly-growing leaves, and of leaves to resist a bloom of browsers, depends greatly on leaf shape.

A larger change in the ecological behavior of the system results from a minor change in the code of line 44. Changing =B to <>SP means that when the browser bites into a leaf edge and then reproduces, the offspring may well appear on the other side of the leaf. This drastic increase in browser mobility will have to be offset by changes in one or more of the set F, G, Z or D.

The resulting stable ecosystem will have leaves that are nibbled to tatters before they get a chance to grow into nice, long leaves. Browser mobility between leaves can be increased by changing the step vectors in line 1100.

Leaf mobility can be introduced by adding a seed dispersal season after line 300, so that some number of single-cell baby leaves are scattered randomly over the screen. Likewise, dispersal of spores might be the proper behavior of browsers who have built up a large population facing dwindling food supplies.

Final Notes

You can follow the "years" better if you increment B after line 240. Then the successive symbols for the browsers become C, D, E and so on down to Z, whereupon you should use an IF to start the cycle over. The program will run OK, but a little illogically, because at any time during the browser-moving season there will be two kinds of browser symbols on the screen, but only one will be recognized by the rest of the program.

On the other hand, if you become really serious about this game, you will want to keep statistics on what is happening, either by storing an internal array, poking variable values onto the screen or outputting to a line printer. Here we part company, since I would rather just fool around with this game and, when the quiet munching of the browsers becomes too tame, open the door to the sabertooths! ■

For fast BASIC programs, put subroutines first and initialization routines last. Don't use decimal constants in inner loops; use variable names instead.

we carry it all....

everything for Commodore
and Atari

Atari® Software

VisiCalc	149
CX4101 Invitation to Programming 1	17
CX4104 Mailing List	17
CX4102 Kingdom	13
CX4103 Statistics	17
CX4105 Blackjack	13
CX4106 Invitation to Programming 2	20
CX4107 Biorhythm	13
CX4108 Hangman	13
CX4109 Graph It	17
CX4111 Space Invader	17
CX4110 Touch Typing	20
CX4115 Mortgage & Loan Analysis	13
CX4116 Personal Fitness Program	13
CX4117 Invitation to Programming 3	20
CX4118 Conversational French	45
CX4119 Conversational German	45
CX4120 Conversational Spanish	45
CX4121 Energy Czar	13
CX4125 Conversational Italian	45
CX8108 Stock Charting	20
CXL4001 Educational System Master	21
CXL4002 Basic Computing Language	46
CXL4003 Assembler Editor	46
CXL4004 Basketball	30
CXL4005 Video Easel	30
CXL4006 Super Breakout	30
CXL4007 Music Composer	45
CXL4009 Chess	30
CXL4010 3-D Tic-Tac-Toe	30
CXL4011 Star Raiders	45
CXL4015 TeleLink	20
Talk & Teach Courseware: CX6001 to CX6017	23

Atari® Peripherals:

400 8K	$ 419
400 16K	449
410 Recorder	62
810 Disk	479
815 Disk	1199
822 Printer	359
825 Printer	759
830 Modem	159
850 Interface Module	164

Atari® Accessories

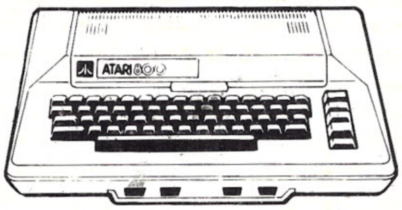
CX852 8K RAM	94
CX853 16K RAM	149
CX70 Light Pen	64
CX30 Paddle	18
CX40 Joystick	18
CX86 Printer Cable	42
CO16345 822 Thermal Printer Paper	5
CAO16087 825 80-col. Printer Ribbon (3/box)	17
Microtek 16K RAM	99
Microtek 32K RAM	179

commodore

VIC-20	$ 279
8096	1890
4032	1080
8032	1499
CBM 4022 Printer	669
CBM 4040 Drive	999
CBM 8050 Drive	1449
CBM C2N Drive	87
PET-IEEE Cable	37
IEEE-IEEE Cable	46

Disks

Maxell Disks	10 for $36
Syncom Disks	10 for 29
Atari Disks	5 for 22



ATARI

Atari® 800™ Personal Computer 32K RAM $759

Software

EBS Accounts Receivable Inventory System	$595
Dr. Daley Mailing List	129
Dr. Daley Inventory	89
OZZ Information System	329
BPI General Ledger	329
Tax Package	399
Dow Jones Portfolio Management	129
Pascal	239
WordPro 3 (40 col.)	186
WordPro 4 (80 col.)	279
WordPro 4 Plus (80 col.)	339

Printers

NEC 5530	2495
Diablo 630	2195
Trendcom 100	299
Starwriter	$1495
Trendcom 200	489
Paper Tiger 445G	769
Paper Tiger 460G	1219
Epson MX-80	499
Tally 8024	1699

Please Call Between 11AM & 6PM (Eastern Standard Time)

No Risk -

No Deposit On Phone Orders -

COD or Credit Card - Shipped Same Day You Call*

Prepaid Orders Receive Free Shipping

(800) 233-8950

* on all in stock units

Computer Mail Order ✓384 **501 E. Third St., Williamsport, PA 17701 (717) 323-7921**

Construct this A/D converter for measurement and control applications.

The Data Grabber

By John Roos

Some of the most fascinating applications of small computers require sensing conditions external to the processor. Data collection, energy management, temperature measurement, industrial applications and home controls all demand that conditions external to the computer be converted to digital form. The numerical representation of the measured quantity is processed by the computer to achieve the desired control or measurement function.

Most low-cost sensors produce an analog output. This is in the form of a voltage, current or resistance change. An analog-to-digital converter converts the analog information (generally a voltage) to a digital word, which is input to the computer.

This eight-bit A/D conversion system, called Data Grabber, will interface to most computers via an eight-bit bidirectional parallel port. The Data Grabber is a complete A/D subsystem consisting of a 16-channel input multiplexer, step-gain amplifier, 3 us A/D and the necessary I/O and control functions to provide a highly versatile device.

Software control of the Data Grabber is completely flexible; any input port can be selected in any order. In addition, the unit provides four input gain settings via a software-controllable step-gain amplifier. Either bipolar or single-ended input modes can be selected under software control.

Prior to discussing the design in detail, it is useful to review representative methods of analog-to-digital conversion.

A/D Conversion

Analog-to-digital conversion is done in a variety of ways. Most trade speed for accuracy or cost. A digital multimeter is a form of A/D converter in which speed is of no consequence as long as the user is not kept waiting more than a few tenths of a second. In computer-based applications, speed becomes important because the quantity being measured might vary rapidly, or because the processor should not be delayed by the A/D.

Most fast conversion schemes use a digital-to-analog converter in some form of feedback servo loop. Fig. 1a shows the basic idea. Here the digital output of an up/down counter controls the D/A converter. The output from the D/A is compared with the analog input signal. The comparator output causes the counter to increment or decrement the digital input to the D/A. Counting continues until the output from the D/A equals the analog input. At this point the comparator causes the counter to stop until the analog input assumes a new value.

This scheme is called a "tracking" analog-to-digital converter. The analog output of the D/A follows the analog input to the comparator. The digital output is taken from the counter and always represents the present value of the analog input signal.

A tracking A/D has some disadvantages. It will follow a single input quite well, provided that the signal does not change value suddenly. If this occurs the counter must traverse many counts to reach the new level. During this period the digital output is incorrect. It is desirable to use a single A/D with a multiplexer so that many input sources can be processed. If these inputs are different in level, then the tracking A/D loses track and must slew to the new level presented by a different input signal when the multiplexer switches.

Consider the case of a ten-bit converter. There are 1024 possible counter states. If one input is at full scale and the next is at zero scale, all 1024 states must be traversed before the digital output correctly represents the new input level. Even if a 1 MHz clock is used, this process will still require 1.024 ms. Thus, the ma-

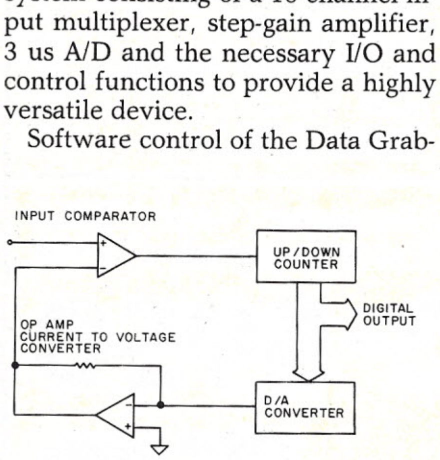


FIGURE 1A

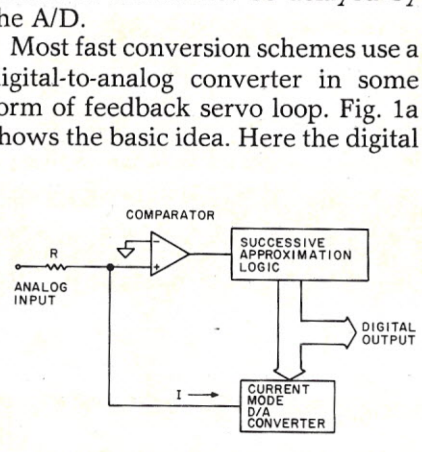


FIGURE 1B

Fig. 1. Two analog-to-digital conversion methods which use feedback from a digital-to-analog converter. The tracking converter (1A) utilizes an up/down counter to instruct the D to A to follow the analog input. A successive approximation approach (1B) is faster because fewer digital states are traversed in forcing the output of the D to A to equal the analog input.

John C. Roos (953 Valley High, Thousand Oaks, CA 91362) is an rf/microwave engineer.

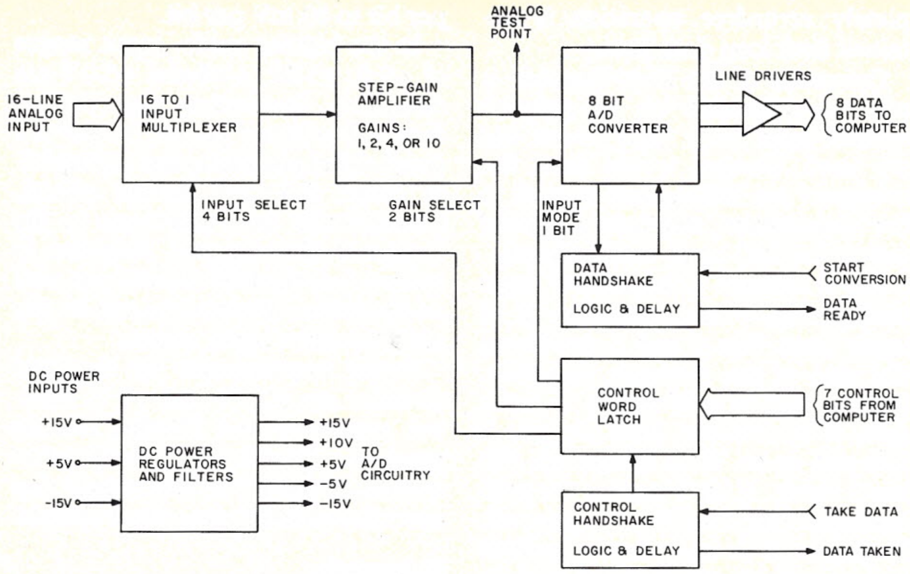


Fig. 2. Data Grabber block diagram. Input signal, amplifier gain and input mode are selected under software control.

jor disadvantage of the tracking A/D is that the conversion time is proportional to the change in input level.

Successive approximation is a technique that overcomes most shortcomings of the tracking A/D. This is a fancy name for trial and error. Fig. 1b shows the circuit arrangement. Like the tracking A/D, a feedback approach is used. A current output D/A is used in preference to a voltage output D/A. This eliminates an operational amplifier, which is often the slowest portion of the circuit.

As the analog input is increased, the circuit acts to sink more current in the D/A, and thus maintain the input voltage to the comparator at zero because of the drop in the input resistor. The comparator output controls the successive approximation logic, which in turn drives the D/A converter.

The trial-and-error conversion process proceeds as follows. First, the most significant bit (MSB) is set to a one, resulting in an output from the D/A of 50 percent of full scale. This is compared with the input. If the input still exceeds the D/A output, the MSB is retained in the one state and the next most significant bit is set to one. If the D/A output had been greater than the input, the MSB would be set to zero prior to setting the next most significant bit to one.

The process continues by again testing the output of the D/A relative to the input, and the state of the second bit is altered accordingly. The third most significant bit is then tested in the same way.

When all bits from MSB to LSB have been tried, the conversion process is complete. It requires $N+1$ trials for a complete conversion for an N bit converter. A ten-bit successive approximation A/D requires 11 trials, as compared to 1024 steps for the tracking A/D approach. Thus, it is about 100 times faster. In the Data Grabber a successive approximation A/D converter is used. An eight-bit conversion requires only 2.8 us.

In selecting an A/D converter the accuracy and required resolution must be considered. They are not the same.

Accuracy refers to the absolute relationship between the digital representation of the input voltage and the true input voltage. On the other hand, resolution refers to the ability to resolve one input level from another. An A/D system can easily have a ten-bit resolution while achieving an accuracy of only 1 percent. Thus, you can see changes in voltage to a resolution of ten bits (0.1 percent), but the voltage is not known to an accuracy of more than 1 percent.

The accuracy will be less than the resolution unless the A/D converter and all analog components which precede it in the input signal path have the necessary precision. This is hard to do; an eight-bit converter requires that all analog errors be held to less than 1/256 or less than 0.5 percent. Gain-setting resistors of such accuracy are both expensive and difficult to obtain. At ten bits and above, you must use extremely precise parts in the input circuitry. If accuracies of that order are required, then a complete commercial A/D subassembly is probably the only way to go.

The application largely determines the required accuracy and resolution. Few sensors are accurate to more than eight bits (<0.5 percent) so for many measurements an eight-bit accuracy and resolution is more than sufficient.

You can use a 12-bit converter to accurately resolve and digitize to a resolution of one part in 4096. But unless you are exceedingly careful, the absolute accuracy will be much less, due to errors in the input circuitry.

Block Diagram

Fig. 2 is a block diagram of the Data Grabber. Analog data is applied to one of 16 signal input lines. A 16-to-1 multiplexer selects the desired signal for conversion. The selected signal is amplified by a step-gain amplifier and then routed to the A/D converter.

The step-gain amplifier permits matching of the input signal dynamic range to the A/D converter so that maximum resolution is maintained. Amplifier gains of 1, 2, 4 or 10 are selectable under software control. Thus, computer-controlled autoranging is possible if required.

A Burr Brown-type ADC82 converter is used. This particular unit features a conversion time of less than 3 us and accommodates both bipolar and unipolar inputs. I used only the 0–10 and ±5 V input ranges. Range extension is provided by the step-gain amplifier. A control line selects either the 0 to +10 V or the ±5 V input ranges.

Data output is via a standard eight-bit parallel port with two handshake lines. One line starts the conversion process; the other signals the computer that valid data is ready. An optional 20 us delay is incorporated into the data handshake logic to permit transients on the data lines to settle prior to sending the data ready signal to the computer.

The parallel interface modules for my H-11 computer have a maximum communications rate of 50,000 words per second. Delaying the data ready signal by 20 us permits the interface to run at its maximum capability while permitting the cross-talk on the data lines to die out. I did this because my application requires a 30-foot cable between the computer and A/D unit. With this delay, completely error-free operation is obtained over long data lines.

Operation of the Data Grabber is software-controlled via an eight-bit control input port. Only seven bits are used. Control words from the computer are latched and applied to the multiplexer, step-gain amplifier and to the A/D input mode control.

Bits 0 through 3 control the input multiplexer, which selects one of 16 input lines. Bits 4 and 5 determine the gain of the step-gain amplifier. Setting both to zero results in unity gain. If bit 4 is one, then the gain is two and so on. Bit 6 selects the A/D input mode. If bit 6 is zero, the positive input (unipolar) mode is selected. Setting bit 6 to one permits positive and negative inputs (bipolar mode).

The combination of software-selectable input ranges and bipolar/unipolar input results in a very versatile A/D conversion system. Table 1 indicates the amplifier gain, input full-scale range and resolution in mV for the various combinations of bits 4, 5 and 6.

The unipolar mode has input full-scale ranges of 0–1, 0–2.5, 0–5 and 0–10 V. Resolution ranges from 3.9 mV per bit to 39 mV per bit.

In the bipolar mode, input ranges of ±0.5, ±1.25, ±2.5 and +5.0 V are obtained. Resolution also ranges from 3.9 to 39 mV per bit; 3.9 mV is a small signal and easily lost in amplifier offset and bias current effects. Considerable care was taken in the design of the step-gain amplifier to ensure that eight-bit accuracy as well as resolution was maintained on the most sensitive input ranges.

Power requirements for the Data Grabber are ±15 V and +5 V. Internal regulation and filtering circuits produce +10 and −5 V, which are required by the input multiplexer and as a reference for some input sensors.

Schematic

Fig. 3 is the complete schematic diagram of the Data Grabber. Analog input signals may be applied to either front panel binding posts or to connectors on the rear panel. The two rear panel input connectors each accommodate eight analog inputs and their respective grounds. A +10 V reference is output from these connectors to bias external devices.

Only eight of the 16 analog input

THE ULTIMATE REFINEMENT.
PL/I FOR YOUR MICRO.

It's standard, it's structured, and it's perfect for commercial applications.

PL/I Subset G. Designed expressly for minicomputers by an ANSI committee of PL/I experts. All the best application programming features, refined to match contemporary programming practices. Supported by the major minicomputer manufacturers, there's no better way to protect your software investment.

PL/I-80™ for Microcomputers. Based on Subset G, PL/I-80 runs under our CP/M© and MP/M™ operating systems on Z-80, 8080 and 8085 based microcomputers.

PL/I-80 Features. Save design time, minimize debugging and maintenance problems, and design high-quality output with picture specifications, structures, arrays, based and pointer variables, and exception processing.

BT-80™. A Record Retrieval System for PL/I-80 applications is available separately to help you manage your large files.

PL/I-80 from Digital Research. Fast. Economical. Refined. With the programming power of minicomputer PL/I. The package includes compiler, run-time library, linkage editor and relocating macro-assembler, plus three comprehensive manuals. All for only $500.

(Domestic USA price only)

Interested in Refinement?

Call or write Digital Research. Experts in refinement.

DIGITAL RESEARCH®

342

U.S.A.
DIGITAL RESEARCH
P.O. Box 579
801 Lighthouse Avenue
Pacific Grove, CA 93950
TWX 910 360 5001

EUROPE
VECTOR INTERNATIONAL
Research Park
B-3030 Leuven
Belgium
32 (16) 20-24-96
Telex 26202 VECTOR

FAR EAST
MICROSOFTWARE ASSOCIATES
102 Plasada
3-16-14 Minami Aoyama
Minato-ku
Tokyo 107, Japan
03-403-2120 Telex 2426875 MSA

Control Bit # 6	5	4	Amplifier Gain	Input Range	Resolution mV/LSB
0	0	0	1	0–10	39.0
0	0	1	2	0–5	20.0
0	1	0	4	0–2.5	10.0
0	1	1	10	0–1.0	3.90
1	0	0	1	±5.0	39.0
1	0	1	2	±2.5	20.0
1	1	0	4	±1.25	10.0
1	1	1	10	±0.50	3.90

Table 1.

channels are available at the front panel, on the theory that any project requiring more than eight inputs is a major undertaking and the rear panel connectors would be preferred.

From the input connectors, the analog voltage is routed to the multiplexer via resistors R1 through R16. These resistors provide overvoltage protection and will limit the current through the input protection diodes within the multiplexers, IC1 and IC2.

The multiplexers are 4051 CMOS transmission gates. Each contains a single-pole, eight-position analog switch. The switches are controlled with a three-bit word which selects the input line. A fourth bit is used as a disable bit. If this bit is a one, no input can be enabled. The disable input is used to expand the multiplexer from eight to 16 input lines by switching one multiplexer off when the other is being used.

The input is selected by connecting the three control bits to both IC1 and IC2. Bit 4, the disable bit, is applied directly to IC1, pin 6. This signal is inverted by IC9D before being sent to the enable input of IC2. When this bit is zero, IC1 is enabled and the inputs from 0 to 7 are active. When bit 4 is set to one, then IC1 is off and IC2 is active. In this state the inputs from 8 to 15 are selected. Open collector buffers IC3 and IC9 interface the CMOS multiplexer to the TTL control latch, IC11.

Input signals applied to the 4051 multiplexers must not exceed the supply voltages. To accommodate the maximum input ranges of 0 to +10 and +5 to −5 V, the 4051 multiplexers are operated from +10 to −5.0 V supplies. The protection resistors limit current in the input diodes of IC1 and IC2. Damage will not occur for inputs of at least 25 V or more.

A higher resistance cannot be used because the step-gain amplifier input bias current passes through the multiplexer and the input protection resistors. The amplifier input impedance is several megohms, but the series resistance must be kept low to preclude input errors due to the bias-current-induced voltage drop in the resistor.

From the multiplexer, the input signal is sent to IC4, the step-gain amplifier. A Harris type HA-2405 programmable amplifier provides four gain settings. Internally, the 2405 consists of four differential input sections, a four-way multiplexer and an output stage. Two digital control lines permit selection of any input section. This versatile chip is a natural for programmable signal processing circuits. Four completely independent feedback networks may be used from the output to each input to provide a wide range of performance variation.

Four gain settings are used: 1, 2, 4 and 10. To obtain a gain of 1, the output, pin 10, is connected directly to pin 5, which is the inverting input of the first section. A gain of 2 is obtained from connection of the output to the inverting input of the second section, pin 7, via the feedback network R28, R29 and R30. R30 is the gain adjustment trimmer.

In similar fashion, the gain of 4 is set with the network R25, R26 and R27, and the gain of 10 is set by R22, R23 and R24. A trimmer is required in each feedback network to set the gain. One percent resistors are used but are of insufficient accuracy for an eight-bit converter. Remember, one LSB equals 1/255, which is a precision of better than 0.5 percent!

Pins 1, 3, 6 and 8 of the 2405 are the noninverting inputs. These are paralleled and connected to the multiplexer outputs.

On the highest gain ranges, the offset voltage of IC4 becomes significant, since one LSB is equivalent to only 3.9 mV. R45 is used in conjunction with R46 and R47 to null the offset of the two highest gain input amplifiers. I got away with a single control because the offsets for two sections of the same chip were about the same. Some 2405s may require a second control.

Capacitor C1 is required to stabilize the 2405 when set up for unity gain. With this value, 100 pF, the amplifier still has a unity gain bandwidth of 2 MHz.

A BNC connector, J3, is provided to connect the output of the step-gain amplifier to external test equipment. J3 is used for gain adjustment. It also may be used to view the analog signal on a scope to detect problems with noise or cross-talk.

From the step-gain amplifier the signal is routed to IC5, the ADC82 analog-to-digital converter. The A/D converter has a number of strapping options for selection of the input signal range. Full-scale ranges may be strapped for 0 to 5, 10 or 20 V in the unipolar mode, and for plus or minus 2.5, 5 or 10 V in the bipolar mode.

In my design, only the 0 to +10 and ±5 V ranges are used because the more sensitive ranges require trimming to obtain one LSB accuracy. The step-gain amplifier is used to obtain more input sensitivity.

This arrangement has two hidden benefits. First the A/D converter has a very low input impedance, only 6.25k. An input buffer is required to raise the input impedance to avoid loading the source and also to preclude excessive voltage drop in the on-resistance of the analog multiplexer. A second benefit is versatility.

Obtaining Parts

Most parts required to construct your own Data Grabber can be purchased from the usual IC vendors. The A/D converter and DG303CJ can be obtained by writing the manufacturers directly and asking for the name of a local distributor. The maker will also sell directly provided you first obtain the total price in advance and then send a check. Here are the addresses:

- Burr Brown Research, International Airport Industrial Park, PO Box 11400, Tuscon, AZ 85734 (602-746-1111).
- Siliconix, 2201 Laurelwood Road, Santa Clara, CA 95054 (408-338-0227).

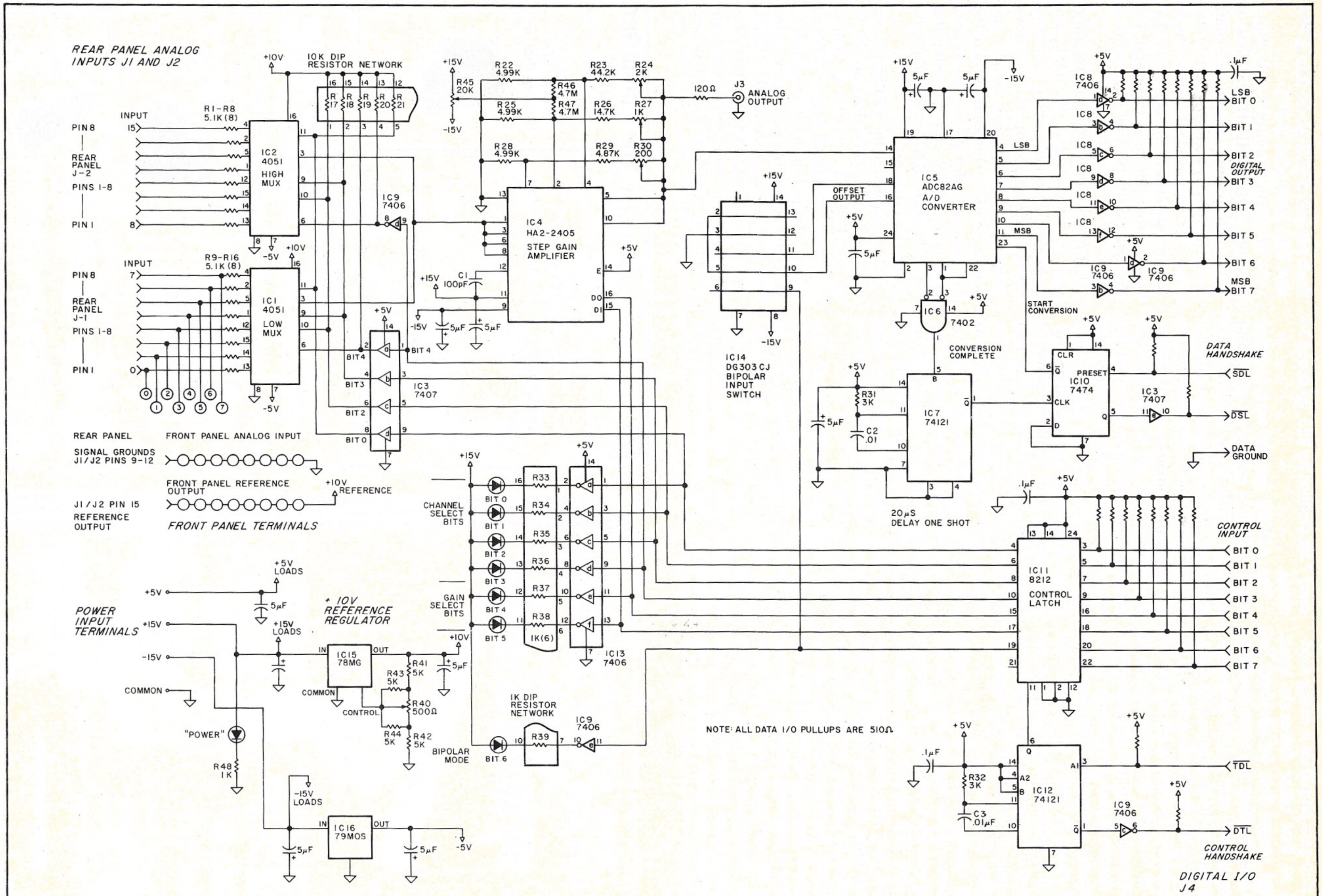


Fig. 3. Data Grabber schematic diagram.

The step-gain amplifier is easily modified so that other full-scale input voltage ranges may be obtained. In the Data Grabber the most sensitive ranges are 0–1 and ± 0.5 V full-scale. More sensitivity is obtained by simply altering the feedback around IC4.

Normally, the A/D is set up for a 0 to +10 V input. Strapping pin 16 to pin 18 connects an internal offset current source to the comparator input for bipolar operation. This offsets the input range so that a −5 V input results in a zero scale digital output. An input of 0.0 V produces a midscale binary output, and +5.0 V produces a full-scale binary output.

The analog switch, IC14, permits selection of the bipolar input mode under software control. A Siliconix type DG 303 switch was used because a very low on-resistance is required. Both sections of the DG 303 are wired in parallel to lower the on-resistance. Each section has an on-resistance of less than 50 ohms. Low resistance is required since the internal offset current source in the A/D has a source impedance of 7.8k. To hold the offset error to less than one LSB, the switch resistance should not exceed 7.8k divided by 255, or about 30 ohms.

The ADC82 has its own internal clock, which is output on pin 1 and strapped to the clock input pin 22. Analog-to-digital conversion is initiated by a high-to-low transition on pin 23. When the conversion is complete, the status output, pin 3, goes low. The start line must be kept low until the conversion is completed. When the status line goes low, the clock is high for one-half clock cycle. A slightly delayed conversion complete signal is obtained by an AND of the clock and the status lines. These are LOW true signals so a 7402 NOR gate, IC6, is used to perform the negative AND.

The conversion complete signal appears at the output of IC6 and 700 ns after the data is valid at the A/D output. This signal actuates the data handshake circuitry, which consists of IC7 and IC10.

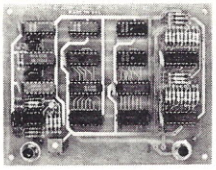
The eight-bit parallel output and handshake lines are designed to interface to a parallel interface module in my Heath H-11 computer. This approach is commonly used, and most computers offer a similar interface capability. The parallel handshake lines are called SDL (send data LOW) and DSL (data sent LOW).

At the completion of a data transmission, SDL is HIGH and DSL is LOW. The computer requests data from the A/D by asserting SDL LOW. The A/D causes DSL to go high, which indicates to the computer that the external device is busy and that the data on the parallel lines is invalid.

For my computer to correctly interpret the busy signal, the DSL line must go high within about 200 ns of the occurrence of SDL. If this is done properly, the external device can take a week to respond. When the A/D has completed the conversion, DSL is set LOW and the computer takes the data. SDL remains high until the next request.

SDL is applied to the PRESET input of IC10, a type 7474 flip-flop. LOW on the PRESET sets the Q output HIGH and the QBAR output LOW. Q is sent to the buffer, IC3 section E, and returned to the computer as the DSL line, which assumes the HIGH, or busy, state. QBAR is sent to the A/D converter as the start conversion command. Analog-to-digital conver-

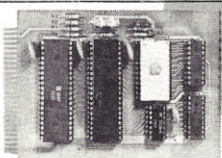
A-D & D-A CONVERTER



JBE A-D & D-A Converter can be used with any system having parallel ports • Interfaces with JBE Parallel I/O Card • D-A conversion time — 5 μs • A-D conversion time — 20 μs • Uses JBE 5V power supply • Parallel inputs & outputs include 8 data bits, strobe lines & latches • Analog inputs & outputs are medium impedance 0 to 5 volt range.

79-287 ASSM. **$79.95**
Bare Board **$39.95** Kit **$59.95**

6502 MICROCOMPUTER



This control computer has: • 1024 bytes RAM (two 2114s) • 2048 bytes EPROM (2716) • Uses one 6522 VIA (comp. doc. incl.) • Interfaces with JBE Solid State Switches & A-D & D-A Converter • Uses JBE 5V power supply • 2716 EPROM available separately (2716 can be programmed with an Apple II & JBE EPROM Programmer & Parallel Interface) • 50 pin connector included in kit & assm.

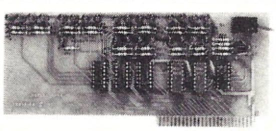
80-153 ASSM. **$110.95**
Bare Board **$49.95** Kit **$ 89.95**

SOLID STATE SWITCH

Your computer can control power to your printer, lights, stereo & any 120VAC appliances up to 720 watts (6 amps at 120VAC). Input 3 to 15VDC • 2-14MA TTL compatible • Isolation — 1500V • Non zero crossing • Comes in 1 or 4 channel version • Includes doc. for interfacing with Dimmer Control.

79-282-1 ASSM. **$13.95**
Bare Board **$6.95** Kit **$10.95**
79-282-4 ASSM. **$49.95**
Bare Board **$24.95** Kit **$39.95**

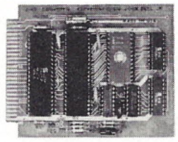
APPLE II DISPLAY BOARD



80-144 ASSM. **$49.95**
Bare Board **$25.95** Kit **$42.95**

• Has run-stop, single step switch • Has 16 address LEDs, 8 data LEDs & 1 RDY LED • All lines are buffered.

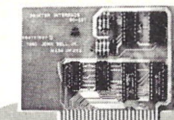
Z80 MICROCOMPUTER



JBE is announcing a single board dedicated computer designed for control functions. It features: • A Z80 Microprocessor software compatible with the Z80, 8080 & 8085 Microprocessors • Uses a Z80 PIO chip for I/O which has 2 independent 8 bit bidirectional peripheral interface ports with handshake & data transfer control • Uses one 2716 EPROM (2K) & two 2114 RAM memories (1K) • Single 5V power supply at 300MA req. • Clock frequency is 2MHz, RC controlled • Board comes with complete doc. • 50 pin connector is included • 2716 EPROM available separately.

80-280 ASSM. **$129.95**
Bare Board **$49.95** Kit **$119.95**

PRINTER INTERFACE



JBE Parallel Printer Interface interfaces your Apple II® to Centronics® compatible printers. This 3" x 4" board features: on board ROM compatible with Integer Basic, Applesoft® and Pascal® • Has one 8 bit parallel latched output port with selectable positive or negative strobe and one bit input selectable for Ready or $\overline{\text{Ready}}$ • Cable and Connectors available separately.

80-297 ASSM.**$79.95**
Kit **$69.95**
4 ft. Std. Dip Jumpers 16 pin **$ 4.25**
Champ Connector **$ 9.95**

POWER SUPPLIES

• Use wall transformers for safety • Protected against short circuit and thermal breakdown.

5 VOLT POWER SUPPLY

Rated at 5V 500MA • Operates JBE A-D & D-A Converter, Z80 & 6502 Microcomputers, 8085 & 8088 Microcomputers.

80-160 ASSM. **$20.95**
Bare Board **$8.95** Kit **$16.95**

± 12 VOLT POWER SUPPLY

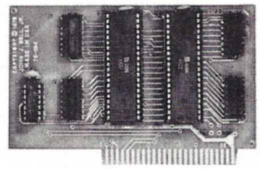
Rated at ± 12V 120MA • Can be used as a single 24V power supply • Ideally suited to OP-AMP experiments.

80-161 ASSM. **$22.95**
Bare Board **$8.95** Kit **$18.95**

ICS

6502	**$9.95**
6522	**$9.95**
Z80	**$9.95**
Z80 PIO	**$9.95**
2716 5V	**$14.95**

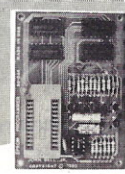
6522 APPLE II INTERFACE



• Interfaces printers, synthesizers, keyboards, JBE A-D & D-A Converter & Solid State Switches • Has handshaking logic, two 6522 VIAs & a 74LS74 for timing. Inputs & outputs are TTL compatible.

79-295 ASSM. **$69.95**
Bare Board **$39.95** Kit **$59.95**

2716 EPROM PROGRAMMER



JBE 2716 EPROM Programmer was designed to program 5V 2716 EPROMS • It can also read 2716s. It interfaces to the Apple II using JBE Parallel I/O Card & four ribbon cable connectors • An LED indicates when power is being applied to the EPROM • A textool zero insertion force socket is used for the EPROM • Comes with complete doc. for writing and reading in the Apple II or Apple II+ • Cables available separately.

80-244 ASSM. **$49.95**
Bare Board **$24.95** Kit **$39.95**

BARE BOARDS

APPLE II EXTENDER BOARD

3½" x 2½". Price includes 50 pin Apple Connector.

80-143 **$12.95**

8085 3 CHIP SYSTEM

State-of-the-art system using an 8085, 8156 & either an 8355 or 8755 • Instruction set 100% upward compatible with 8080A.

Bare Board **$24.95**

8088 5 CHIP SYSTEM

An 8086 family microcomputer system using an 8088 CPU, 8284, 8155, 8755A & an 8185.

Bare Board **$29.95**

CRT CONTROLLER

This intelligent CRT Controller is based on an 8085A Microprocessor & an 8275 Integrated CRT Controller. It features: • 25 lines, 80 characters/line • 5x7 dot matrix • Upper case only • Two 2716s • Serial Interface RS232 & TTL • Baud rates of 110, 150, 300, 600, 1200, 2400, 4800 & 9600 • Keyboard scanning system • Req's. unencoded keyboard • Uses +5V & ± 12V power supplies.

Bare Board **$39.95**

50 pin connector **$5.95**
STD. Dip Jumpers 16 Pin, 2 ft. **$4.25**

JOHN BELL ENGINEERING ✓99

MC

ALL PRODUCTS ARE AVAILABLE FROM **JOHN BELL ENGINEERING** • P.O. BOX 338 REDWOOD CITY, CA 94064 • ADD 6% SALES TAX IN CALIFORNIA • ADD 5% SHIPPING & HANDLING

(415) 367-1137

VISA

✓See List of Advertisers on page 210

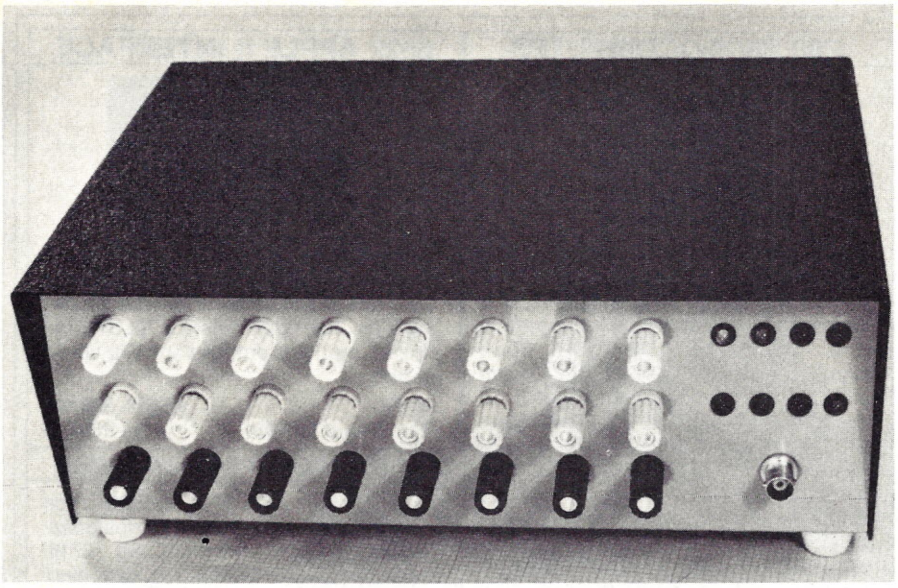


Photo 1. The Data Grabber is housed in a small instrument case. Front panel connections include eight inputs, with ground returns, and eight reference voltage outputs. BNC connector, J3 at lower right, is the analog output test point. Seven LEDs indicate the command status. An eighth LED indicates power on.

sion is initiated. When the conversion is complete, valid data appears on the parallel output lines.

IC6 outputs a delayed (by about 700 ns) conversion complete signal to IC7, a 74121 monostable. IC7 is used to delay the sending of the DSL signal so that the transients on the data lines have time to die out. This is not strictly necessary, but it does reduce the errors on long lines. When IC7 times out, its QBAR output returns to the HIGH state and clocks IC10. The Q output of IC10 is clocked LOW and returns the DSL line to the LOW condition, which indicates to the computer that it should take the data.

Data from the A/D is output in a complementary binary format. 7406 inverting buffers, IC8 and IC9, invert the data and drive the output lines to the computer. After this inversion the data is in normal binary format; 00000000 is zero scale and 11111111 is full scale for the unipolar input ranges. The code is offset when the bipolar input mode is used. 00000000 is output for negative full-scale inputs; 11111111 is transmitted for positive full-scale inputs. An input of 0.0 V results in a midscale binary output, or 10000000.

The line drivers use 510 ohm pull-up resistors to reduce the effects of capacitive coupling between lines. Low impedance loads, combined with the intentional 20 us handshake delay, permit operation with at least a 30-foot cable between the A/D converter and the computer.

A similar eight-bit port permits the computer to control the Data Grabber. Only the seven least significant bits are used. Two handshake lines, TDL (Take Data LOW) and DTL (Data Taken LOW), control the command word transmissions.

A command from the computer is sent to IC11, an 8212 eight-bit latch. Then the TDL line is asserted LOW by the computer parallel interface. IC12, a 74121 monostable, is set by this signal. The Q output of IC12 goes HIGH, which in turn causes the 8212 latch to become transparent to the input data. The QBAR output of the 74121 is inverted and returned to the parallel interface as the DTL signal. This line assumes the HIGH, or busy, condition.

IC12 then times out and returns pin 11 of the latch LOW. This latches the command word. At the same time QBAR of IC12 returns HIGH and the DTL line is returned LOW by the inverting output buffer IC9 section C. Assertion of DTL LOW informs the computer that the command word has been received by the Data Grabber.

From the command latch, the seven control bits are routed to the multiplexer, step-gain amplifier, input mode switch and LED display drivers.

Bits 0–3 select the input channel. They are routed to the input multiplexer via IC3 and IC9, which interface the TTL signals to the CMOS multiplexers.

Bits 4 and 5 control the step-gain amplifier. They connect directly to input selector lines D0 and D1 of the 2405 step-gain amplifier.

The last control bit, bit 6, connects to the control port of the mode selection switch, IC14. Bit 7 is not used.

A front panel display indicates the current command status. Seven LEDs are used to indicate the state of each bit in the command word. This feature is useful when debugging the control software. By single-stepping the program through the programmed control sequence, proper operation can be verified. The LEDs are driven by 7406 inverting buffers, IC13 and one section of IC9.

Power for the Data Grabber is supplied from external ±15 and +5 V supplies. Regulators IC15 and IC16 develop +10 and −5 V, which are used internally. The +10 V supply powers the input multiplexer and is also output as a precision reference.

A reference voltage is useful to bias sensors used with the A/D converter. Thermistors or other resistive devices require stable bias for operation. The +10 V supply is routed to the front panel binding posts and to the rear panel input connectors for this purpose.

A 78MG adjustable regulator is used for the +10 volt reference supply. While not as good as a super reference diode, its stability with temperature is adequate and the cost is much lower. Temperature stability of the reference supply should be much better than one LSB because drift in the reference will directly affect measurement accuracy. Of course, this is most critical on the more sensitive input ranges. Let's see how the reference compares to one LSB.

A temperature coefficient of 0.4 mV per degree Centigrade is specified for the 78MG regulator. Thus, its output can be expected to remain within ±4 mV for a ten-degree increase or decrease in ambient temperature. One LSB equals 39 mV on the 0–10 V input range and 3.9 mV on the 0–1 V range. So under most conditions the reference will contribute less than one LSB error.

Precision resistors, R41 through R44, are used with trimmer, R40, to set the reference supply output. Use 1 percent metal film resistors if obtainable. R43 and R49 are included to prevent an overvoltage output from IC15 should the pot wiper go open. IC15 can supply over 500 mA. Overvoltage damage to the multiplexer could result in an excessive input to the step-gain amplifier chip. The

CMOS multiplexers are inexpensive, but the 2405 is not.

Construction

I constructed the Data Grabber on a 5×11 inch piece of one-sixteenth inch copper clad PC card. The card is housed in a 4.5×8×11 inch cabinet (Photo 1).

While the layout is not terribly critical, the circuit does handle signals as small as a few mV. Use of a ground-plane is mandatory to prevent pickup of hum and noise. One reason for not including a power supply in the same cabinet was to provide isolation from an obvious source of magnetic interference.

I arranged the circuit components on the board for maximum isolation of the analog and digital portions. The analog inputs are at the lower right of the circuit card (Photo 2). Sixteen input protection resistors are mounted on Teflon standoffs. Above these are the multiplexers, IC1 and IC2, and the 2405 step-gain amplifier, IC5. All of the gain setting and offset adjustments and their associated fixed resistors are at the extreme upper right of the card. Orientation of the trimmer controls to the right of the card permits adjustment by removal of the top of the chassis. It is not necessary to remove the card from the box.

The A/D converter is located at the left of the analog circuits. To its right is the input mode switch, IC14. At the left of the A/D are ICs 8, 9, 6, 7, 10 and 11. Starting from the left of the card, the next row contains a 1k DIP resistor network, IC13, IC12, the timing components for the two one shots, IC3 and the 10k resistor network. This last item is adjacent to the input multiplexer IC2.

Both voltage regulators are at the lower-left edge of the card.

At the upper-left edge of the card are the 510-ohm pull-up resistors for the TTL I/O lines. Twenty resistors are used. DIP resistor networks could have been used, but in any event, a terminal is required to connect the 30-gauge wire-wrap wire under the card to the 22-gauge leads to the connectors. In addition, the terminal provides a ready-made test point, so use of discrete resistors is convenient.

All ICs are mounted in wire-wrap sockets, and wire-wrap interconnections are used except where a lead terminates at a Teflon feed-through. These connections are soldered.

On the front panel are located the input connections, reference output terminals, the analog output BNC connector and the LEDs for the status display. Rear panel connectors include the two analog input connectors, Digital I/O and a power connector. A set of binding posts, which duplicates the power input connections, is also provided.

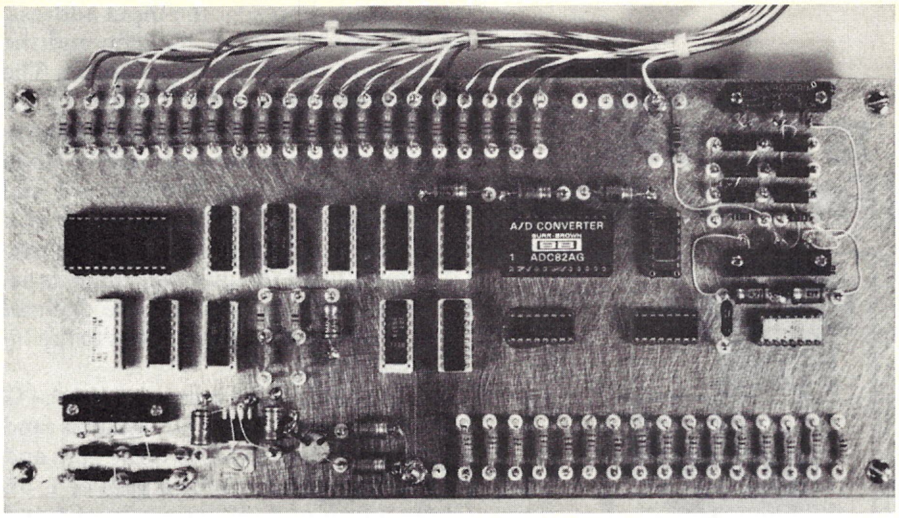


Photo 2. A single circuit card holds all components. Wire-wrap construction is used. A copper ground plane is required to prevent cross-talk and noise pickup.

```
100 REM THIS PROGRAM SENDS A SINGLE COMMAND TO THE A TO D
110 REM CONVERTER. THEN IT REPEATEDLY READS THE DIGITAL OUTPUT
120 REM USE TO TEST THE -DATA GRABBER-
200 PRINT "TYPE COMMAND NUMBER 0-127 ";
210 INPUT N
300 REM COMMAND IS POKED TO A TO D VIA PARALLEL INTERFACE
310 REM 65388 IS ADDRESS OF TRANSMITTER STATUS REGISTER
320 REM 65390 IS ADDRESS OF TRANSMITTER DATA BUFFER
330 POKE 65388,4
340 POKE 54390,N
350 REM NUMBER N SENT TO DATA GRABBER
500 REM READ DIGITAL OUTPUT IN A LOOP
510 REM 65384 IS RECEIVER STATUS REGISTER ADDRESS
520 REM 65386 IS RECEIVER INPUT DATA BUFFER
530 REM BEGIN LOOP
540 REM SET UP INTERFACE TO READ IN ONE BYTE
550 POKE 65384,1
560 POKE 65384,0
570 REM READ BYTE AND STORE IN A
580 A=PEEK(65386)
590 REM NEXT SET TO +10 VOLT INPUT FULL SCALE
600 B=10*(A/255)
700 PRINT B;
710 REM SCALED INPUT IS PRINTED ON CRT AS VOLTAGE 0-10 VOLTS
720 REM END OF ONE INPUT LOOP
730 GO TO 530
1000 REM HEATH H-11 PAPER TAPE BASIC. J. C. ROOS JUNE 80
```

Listing 1. A BASIC program for sending command words to the Data Grabber during checkout.

Testing

Testing of the A/D converter should be done after assembly and wiring of the card and prior to installing it in the chassis.

Test the dc regulators and wiring first. Adjust the +10 V supply for an output of exactly 10.000 V. Do not install the A/D converter until you have verified that everything is OK. Check that not only the dc voltages supplied to the A/D are correct, but also that the digital outputs are not shorted.

Write a short program to drive the Data Grabber. This can be pretty simple. All it must do is send a control word to the unit and then continually interrogate the digital output. Listing 1 is the program I used with my H-11 computer. This program permits you to send a command where the number N is the decimal representation of the binary command word.

Remember, the binary command is divided into subsections. The first

four bits represent the input address, the next two select the gain and the last one selects the input mode. The number which BASIC pokes to the A/D is the sum of the decimal equivalents of these binary bits. If N = 0 then input 0 is selected, the gain is 1 and the unipolar mode is selected. If N = 16, then the input 0 is selected, the gain is 2 and the unipolar mode is selected. Adding 64 to N (the decimal equivalent of bit 6) switches to the bipolar mode.

To state it more formally, $N = A + G + M$, where A is the address and ranges from 0 to 15. G is the gain and has four values: 0 for a gain of 1, 16 for a gain of 2, 32 for a gain of 4 and 48 for a gain of 10. M, the mode select bit equivalent, is 0 for unipolar inputs and 64 for bipolar signals. In a running program any combination of these may be sent to the A/D in any sequence.

Your control program may be tested by sending commands to the A/D converter and observing the LED display. Once this is working, the step-gain amplifier can be calibrated.

The first step is to null the amplifier offset. Have the computer command the A/D for a gain of 10 with input 0 selected. Ground input 0. Then adjust R45 for a zero voltage output from J3. If you cannot null the output exactly, at least obtain a voltage less than 20 mV. This is less than one-half of one LSB.

Next, command a gain of 4 with input 0 selected. Verify that the offset remains less than 20 mV for this input amplifier. If it does, proceed. If it does not, then the input amplifiers of the 2405 are sufficiently different that a second pot will have to be installed to control the voltage applied to R47.

Once the offsets are nulled for the high gain inputs, the amplifier gains are set up. A good digital multimeter and a variable voltage source are required. The DMM should have a specified accuracy and resolution of at least 0.1 percent or better. Remember, the Data Grabber is good for 0.5 percent. You need to calibrate it with something much better than that if the calibration device is not to contribute to the ultimate measurement error.

Program the Data Grabber for a gain of 2 with input 0 selected. Apply an input of exactly 5.00 V dc. Set R30 for a +10.00 V output from J3.

Next, program a gain of 4 with input 0 selected. Apply exactly 2.500 V to input 0. Adjust R27 for a 10.00 V output at J3. Finally, program a gain of 10. Repeat the adjustment process with an input of 1.000 V and set R24 for a 10.00 V output.

Repeatedly check both the input and output to ensure that nothing drifts during the measurement. Your voltage source should have an output impedance of less than 1k to prevent bias current errors.

Verify that all voltages and loads presented to the ADC 82 are correct and insert it into its socket with the power off. Using your control program and the computer, test for proper operation by selecting the various inputs, gains and modes. Connect the DMM to the variable voltage source and ensure that the number displayed on the computer correctly tracks the input voltage.

It is useful to examine the parallel data lines to the computer. They should be free of any ringing and overshoot when the DSL line goes low.

Conclusion

Construction of the Data Grabber will provide a useful building block for a variety of computer-based projects. The design permits considerable freedom for modification to individual requirements. Here are some ideas.

Reduction of the timing period of the delay one shots will enable the circuit to operate much faster. With short lines from the A/D to the computer, the delays may be set to 500 ns or less. Then your problem is getting the software to cycle at a 4 us rate. Generally, the computer software, not the A/D converter, will set the operating speed. If the unit is controlled with assembly-language routines, it may be useful to reduce the delay times. If you drive the unit with software in BASIC, it will be much faster than the software.

As only seven of the eight possible control bits are used, it is easy to expand the input multiplexer. An additional bit and some gates for decoding would permit up to 32 input channels.

In writing this article I have tried to explain the reasons for most features of the design. Evaporation of the basic accuracy of the A/D converter is a subtle problem and one which is not often discussed. With the information presented, you should be able to use my design as is, or as a starting point for your own creation. ■

This software approach, which uses PET's ability to call another program into memory, is not without its problems; the author helps you avoid these pitfalls.

Expand PET Memory

By Gary L. Ratliff

It is possible to get added memory for larger application programs by using the PET's ability to call into memory another program. This ability is also possible with several popular computers which have Microsoft BASIC.

This technique has its pitfalls, the most obvious of which is that you can crash the system by having a small program call and try to execute a much larger program. My technique for circumventing these problems is designed for the PET computer, but should also help owners of other systems find similar routines for their computers.

Variable Storage Area

To design a system of programs which will call into memory and execute subprograms, you must know how to pass parameters from one program to another. To understand this, you have to know how variables are stored internally in the PET. Listing 1 gives a dump of the contents of the variable table, in the proper format for easy interpretation by the user. It also dumps the contents of the program and variable table in screens which contain 32 bytes each.

In Listing 1, lines 10–40 set up examples of the four types of simple variables: user functions, strings, floating point constants and integer constants. In line 110, I is used as both the format control variable and

Address correspondence to Gary L. Ratliff, 1033 Hattiesburg Drive, Magee, MS 39111.


```
10 DEFFNA(X)=INT(X/256):DEFFNB(X)=(X/256-FNA(X))*256
20 A$="ABC":B$="CDE":B$=B$+""
30 A=1:B=-1:AB=281:BA=-281
40 A%=1:B%=-1:AB%=281:BA%=-281
100 GOSUB300
105 FORJ=PEEK(124)+256*PEEK(125)TOPEEK(126)+256*PEEK(127)STEP7:PRINT
110 FORI=0TO36STEP6
115 PRINTTAB(I)STR$(PEEK(J+INT(I/6)));
120 NEXT:IFPEEK(225)=131THENM=1:GOSUB500
130 PRINT:NEXT:M=2:PRINT"END OF TABLE":GOSUB500
200 FORX=1024TOPEEK(126)+256*PEEK(127)STEP2
205 PRINTTAB(0)PEEK(X);TAB(7)FNB(X);TAB(14)FNA(X);TAB(22)PEEK(X+1);TAB(29)FNB(X
+1);TAB(36)FNA(X+1)
210 IFPEEK(225)=131THENGOSUB500
211 REM GO
215 NEXT:PRINT"END OF TABLE":END
300 L=L+1
301 PRINT"BASIC VARIABLE TABLE":PRINT"SCREEN #";L:PRINT
305 PRINT"6 5 4 3 2 1 0":RETURN
400 P=P+1:IFX>256*PEEK(125)THEN402
401 PRINT"PROGRAM AND VARIABLE STORAGE":PRINT"SCREEN #";P:GOTO410
402 PRINT"PROGRAM AND VARIABLE STORAGE":PRINT"SCREEN #";P
410 PRINT"DATA LOC. PAGE DATA LOC. PAGE":RETURN
500 PRINT:PRINT"HIT ANY KEY TO CONTINUE"
505 GETA$:IFA$=""THEN505
510 ON M GOSUB300,400
515 RETURN
```

Listing 1. Variable storage area formatted dump (old ROMs). Line 211 is entered as 211 REM k. On the old PETs it will stop the screen listing when a LIST is given but have no effect during a run. 211 REM l will do the same thing for new PETs but those with printers will find this unnecessary as it will also stop a printer.

Listing 1a. Variable storage formatted dump (new ROMs).

```
10 DEFFNA(X)=INT(X/256):DEFFNB(X)=(X/256-FNA(X))*256
20 A$="ABC":B$="CDE":B$=B$+""
30 A=1:B=-1:AB=281:BA=-281
40 A%=1:B%=-1:AB%=281:BA%=-281
100 GOSUB300
105 FORJ=PEEK(42)+256*PEEK(43)TOPEEK(44)+256*PEEK(45)STEP7:PRINT
110 FORI=0TO36STEP6
115 PRINTTAB(I)STR$(PEEK(J+INT(I/6)));
120 NEXT:IFPEEK(197)=131THENM=1:GOSUB500
130 PRINT:NEXT:M=2:PRINT"END OF TABLE":GOSUB500
200 FORX=1024TOPEEK(44)+256*PEEK(45)STEP2
205 PRINTTAB(0)PEEK(X);TAB(7)FNB(X);TAB(14)FNA(X);TAB(22)PEEK(X+1);TAB(29)FNB(X
+1);TAB(36)FNA(X+1)
210 IFPEEK(197)=131THENGOSUB500
215 NEXT:PRINT"END OF TABLE":END
300 L=L+1
301 PRINT"BASIC VARIABLE TABLE":PRINT"SCREEN #";L:PRINT
305 PRINT"6 5 4 3 2 1 0":RETURN
400 P=P+1:IFX>256*PEEK(43)THEN402
401 PRINT"PROGRAM AND VARIABLE STORAGE":PRINT"SCREEN #";P:GOTO410
402 PRINT"PROGRAM AND VARIABLE STORAGE":PRINT"SCREEN #";P
```

More

Listing 1a continued.

```
410 PRINT"DATA▶▶LOC.▶▶PAGE▶▶DATA▶▶▶LOC.▶▶PAGE":RETURN
500 PRINT:PRINT"■HIT ANY KEY TO CONTINUE"
505 GETA$:IFA$=""THEN505
510 ON M GOSUB300,400
515 RETURN


LISTING 1A WITH PAGINATION ON LISTING ADDED VIA LINE 211


10 DEFFNA(X)=INT(X/256):DEFFNB(X)=(X/256-FNA(X))*256
20 A$="ABC":B$="CDE":B$=B$+""
30 A=1:B=-1:AB=281:BA=-281
40 A%=1:B%=-1:AB%=281:BA%=-281
100 GOSUB300
105 FORJ=PEEK(42)+256*PEEK(43)TOPEEK(44)+256*PEEK(45)STEP7:PRINT
110 FORI=0TO36STEP6
115 PRINTTAB(I)STR$(PEEK(J+INT(I/6)));
120 NEXT:IFPEEK(197)=131THENM=1:GOSUB500
130 PRINT:NEXT:M=2:PRINT"■END OF TABLE":GOSUB500
200 FORX=1024TOPEEK(44)+256*PEEK(45)STEP2
205 PRINTTAB(0)PEEK(X);TAB(7)FNB(X);TAB(14)FNA(X);TAB(22)PEEK(X+1);TAB(29)FNB(X
+1);TAB(36)FNA(X+1)
210 IFPEEK(197)=131THENGOSUB500
211 REM

215 NEXT:PRINT"■END OF TABLE":END
300 L=L+1
301 PRINT"■▶▶▶▶▶▶▶▶▶▶BASIC VARIABLE TABLE":PRINT"SCREEN #";L:PRINT
305 PRINT"▶6▶▶▶▶▶5▶▶▶▶▶4 ▶▶▶▶3▶▶▶▶▶2▶▶▶▶▶1▶▶▶▶▶0":RETURN
400 P=P+1:IFX>256*PEEK(43)THEN402
401 PRINT"■▶▶▶▶▶▶■PROGRAM■ AND VARIABLE STORAGE":PRINT"SCREEN #";P:GOTO410
402 PRINT"■▶▶▶▶▶▶PROGRAM AND ■VARIABLE■ STORAGE":PRINT"SCREEN #";P
410 PRINT"DATA▶▶LOC.▶▶PAGE▶▶DATA▶▶▶LOC.▶▶PAGE":RETURN
500 PRINT:PRINT"■HIT ANY KEY TO CONTINUE"
505 GETA$:IFA$=""THEN505
510 ON M GOSUB300,400
515 RETURN
```

the index pointer indicating the byte to be printed to the screen. The STR$ function in line 115 prevents the PET from printing a trailing blank space should a numeric value instead of the equivalent string value be printed.

If bit 7 is on, the ASCII value of the variable is ORed with 128 to produce its ultimate code. For example, the variable AA may represent a function name, a string, a floating point constant or an integer constant.

ASCII code for A is 65. If this value is ORed with 128, it becomes 193. In Table 1, you can see that the designation 65 65 means the floating point value AA; the designation 65 193 means the string value AA; the designation 193 65 means the function AA; and the designation 193 193 means the integer value AA.

Next, let's examine the output of Listing 1 to see how the PET deals with each type of variable.

Listing 1 also has two automatic pagination features. In the old PET, memory location 225 contains the page number printed to the screen


```
              BASIC VARIABLE TABLE
SCREEN # 1
 6     5     4     3     2     1     0
 193   0     12    4     7     9     181    FNA
 88    0     0     0     0     0     0      X
 194   0     28    4     7     9     40     FNB
 65    128   3     54    4     0     0      A$
 66    128   3     78    126   0     0      B$
 65    0     129   0     0     0     0      A
 66    0     129   128   0     0     0      B
 65    66    137   12    128   0     0      AB
HIT ANY KEY TO CONTINUE
```

Fig. 1. Output of Listing 1b, screen 1.

```
              BASIC VARIABLE TABLE
SCREEN # 2
 6     5     4     3     2     1     0      BA
 66    65    137   140   128   0     0
 193   128   0     1     0     0     0      A%
 194   128   255   255   0     0     0      B%
 193   194   1     25    0     0     0      AB%
 194   193   254   231   0     0     0      BA%
 67    82    144   0     0     0     0      CRT
 87    68    134   32    0     0     0      WD
 76    0     130   0     0     0     0      L
HIT ANY KEY TO CONTINUE
```

Fig. 2. Output of Listing 1b, screen 2.

```
              BASIC VARIABLE TABLE
SCREEN # 3
 6     5     4     3     2     1     0
 74    0     140   22    224   0     0      J
 73    0     132   16    0     0     0      I
END OF TABLE
HIT ANY KEY TO CONTINUE
```

Fig. 3. Output of Listing 1b, screen 3.

```
          PROGRAM AND VARIABLE STORAGE
SCREEN # 1
DATA   LOC.    PAGE   DATA    LOC.    PAGE
 0      0       4      46      1       4
 4      2       4      0       3       4
 0      4       4      150     5       4
 165    6       4      65      7       4
 40     8       4      88      9       4
 41     10      4      178     11      4
 181    12      4      40      13      4
 88     14      4      123     15      4
 50     16      4      53      17      4
 54     18      4      41      19      4
 58     20      4      150     21      4
 165    22      4      66      23      4
 40     24      4      88      25      4
 41     26      4      178     27      4
 40     28      4      88      29      4
 173    30      4      50      31      4

HIT ANY KEY TO CONTINUE

          PROGRAM AND VARIABLE STORAGE
SCREEN # 3
DATA   LOC.    PAGE   DATA    LOC.    PAGE
 68     64      4      69      65      4
 34     66      4      58      67      4
 66     68      4      36      69      4
 178    70      4      65      71      4
 36     72      4      170     73      4
 34     74      4      34      75      4
 0      76      4      165     77      4
 40     78      4      30      79      4
 0      80      4      63      81      4
 178    82      4      40      83      4
 58     84      4      65      85      4
 178    86      4      121     87      4
 49     88      4      58      89      4
 65     90      4      66      91      4
 178    92      4      50      93      4
 56     94      4      49      95      4

HIT ANY KEY TO CONTINUE

          PROGRAM AND VARIABLE STORAGE
SCREEN # 2
DATA   LOC.    PAGE   DATA    LOC.    PAGE
 53     32      4      54      33      4
 121    34      4      165     35      4
 65     36      4      40      37      4
 88     38      4      41      39      4
 41     40      4      122     41      4
 50     42      4      53      43      4
 54     44      4      0       45      4
 72     46      4      40      47      4
 28     48      4      0       49      4
 65     50      4      36      51      4
 178    52      4      34      53      4
 65     54      4      66      55      4
 67     56      4      34      57      4
 58     58      4      65      59      4
 36     60      4      178     61      4
 34     62      4      67      63      4

HIT ANY KEY TO CONTINUE
```

First portion of remainder of printed run of Listing 1b for the program and variable storage area.

(location 197 for new PETs). After a clear screen this value is 128 (256 × 128 = 32768, the start of screen memory). When this value reaches 131, 20 lines of text have been printed and subroutine 500 stops further printing.

Line 211 causes automatic pagination when the program is listed to the screen. When shifted characters are entered following a remark statement, they will be printed as a keyword when the program is listed. Using a shifted K after a REM forces an interrupt when the program is listed. (On new PETs you use a shifted L.) This line causes the message

? SYNTAX ERROR

	Byte 6	Byte 5	
	0	0	Floating Point
	0	1	String
	1	0	Function
	1	1	Integer

Test for bit 7 ON, which means is it equal to 128 or is it more than 128?

Table 1. Truth table for variable discrimination.

READY

to be printed to the screen, and the listing will stop. To view page two of the program listing, enter the direct command LIST 212 and press return.

Line 205 is entered in PET shorthand by typing:

205?Ta0)Pe(X);Ta7)FNB(X);Ta14)FNA(X);Ta22)Pe(X + 1);Ta29)FNB(X + 1);Ta36)FNA(X + 1)

The output of the portion of Listing 1 that prints the variable table is given in Figs. 1 and 2. Table 1 shows how bytes 5 and 6 are used to determine which type of variable is being referenced. The value 0 indicates that bit 7 is off, while the value 1 indicates that this bit is on.

Line 10 introduces two user-defined functions which share the same "dummy" argument. Columns 6 and 5 determine that this is a function. Columns 4 and 3 point to the text of the program where this value is defined. Columns 2 and 1 point to the location in variable storage which contains the dummy argument to be used by this function. Column 0 contains the code of the first word of the function definition. Thus, 181 is the code for the INT function of FNA, and 40 is the ASCII code for "(", which is the start of FNB. The dummy argument of the function is entered and initialized at zero. Since functions A and B share the same argument, both contain the pointers 83 and 7.

Line 20 introduces two strings and performs an operation on string B. Columns 3 and 2 point to the location of the string in memory. Column 4 tells the length of the string. Note that operating on string B moves it to storage in high memory.

Line 30 introduces floating-point constants. These numbers are represented in excess-128-binary floating-point notation in columns 4 to 0. (This system of notation is given full treatment in Mark Zimmerman's article "Snooping with Your PET," *People's Computers*, July–August 1978).

Line 40 introduces integer constants. These numbers are represented in columns 4 and 3. Column 4 is the high byte. If the number is negative, it is represented in two's complement form.

Use the output of Listing 1 to establish rules for passing parameters from one program to another. Simple numeric variables or arrays may be passed directly from one program to another in the variable table storage area of this program. Strings may be passed by first operating upon them

See List of Advertisers on page 210

to move them to high memory, and readjusting their pointers to this space.

The null string is to the concatenation operation as zero is to addition: it is the identity element, and its use allows strings to be passed unaltered. A complete table of string information may be passed from one program to the other programs in the system by operating on the string when it is read or input into memory. For example,

10 FOR I = 0 TO 255 : READ A$(I): A$(I) = A$(I) + " ":NEXT

would pass a table contained in data statements into an area of high memory. Then, if the next program module needs this information, it is available without repeating the data statements.

User-defined functions may not be passed from one program to another; they must be repeated within the text of each subprogram that requires their use.

Size Incompatibility

The second pitfall in using the load feature to create longer programs is size incompatibility. Consider C. Ascolillo's article "Cover Up," (August 1979 *Kilobaud Microcomputing*). This series of five programs could easily be modified to run serially by changing the END statement to a LOAD statement. However, the Menu program is not size-compatible with the other programs in this series.

There are two methods of correcting the size-incompatibility problem; which you use depends on whether you take the *Commodore User's Manual* literally or figuratively. I first used the program call in my programs to create a machine-language system. This system contained a 250-byte machine-language routine loader, which called in a 4K disassembler program, which called in the Monitor.

I overcame the size-incompatibility problem by making the first program longer than 4K. Additional lines were added which stated

27 REM THE PURPOSE OF THIS LINE IS TO USE MORE MEMORY.

I used the screen editor to change the line number, and this short program was soon over 4K long. But, while the very short program loaded into memory in seconds, it now requires minutes to load.

The small program that attempts to call a larger program into memory has one of the characteristics of an epic—its variable table now starts *in medias res.* The program called loads perfectly. To prove this, let the first step of a program be STOP. Next, save this altered program. Write a short program to act as a loading routine: 10 LOAD will do the trick. Next, run the program.

When you type list, you will note that the entire program is listed. However, when you determine the start of the variable table, you find it on page 4, location 10 of memory. This is just ten bytes into the text of the newly-loaded program.

The feature which allows the use of the command LOAD from within the program to create an overlay, by not automatically adjusting the variable storage area to the size of the incoming text, is what makes passing parameters possible. This also causes the incoming program to self-destruct if the calling program is smaller than the called program.

The solution is for the short program to redefine its size so that, in effect, it has the same length as the largest program that will be used by the system. Before saving the lead program of your system, add the following line to the text:

0 REM POKE 124,XX: POKE 125,YY: RUN ZZ

For the new PETs add:

0 REM POKE 42,XX: POKE 43,YY: RUN ZZ

If the first line of the program is 10, then the value 10 is substituted for ZZ.

The other values should each contain three bytes. As each of the subprograms is written and tested, its size should be determined by peeking locations 124 and 125 (42 and 43 on the new PETs). When the system of programs is finished, the value obtained for the largest program within the system should be entered into the lead program, and the REM portion deleted.

Line 0 is a dummy line; its purpose is to reserve space on the cassette tape so that correct line will not overwrite the file header of the program, which is the second in your series of programs.

For example, to correct the machine-language system so that the first program loads as a 250-byte program in a matter of seconds, and yet appears to the PET to be the same size as the disassembler program it calls, the only modification would be to add line 0:

0 POKE 124,154: POKE 125,24: RUN1.

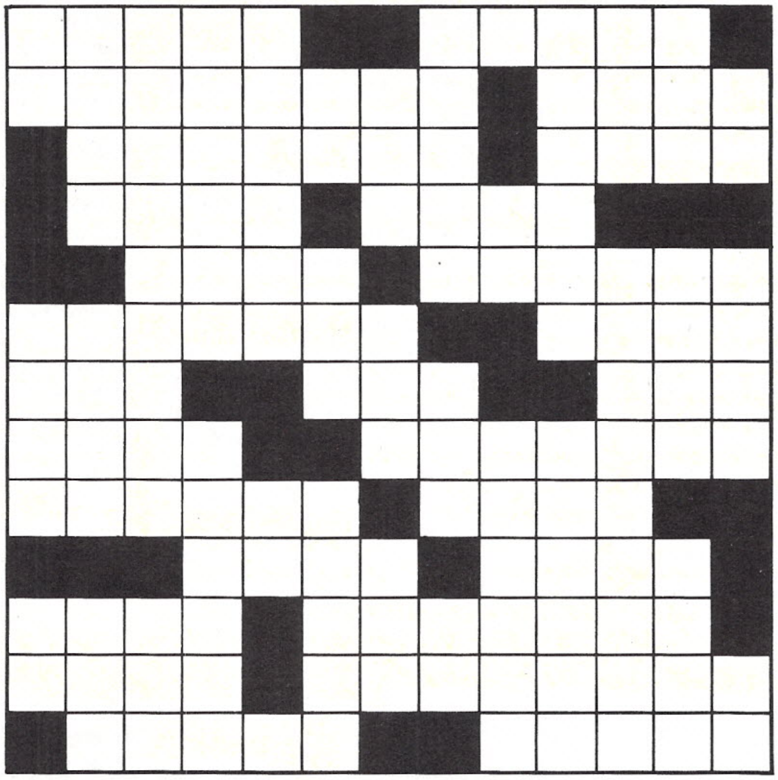
Why is the directed run important? When the natural size of the

```
10 DEFFNA(X)=INT(X/256):DEFFNB(X)=(X/256-FNA(X))*256
20 A$="ABC":B$="CDE":B$=B$+""
30 A=1:B=-1:AB=281:BA=-281
40 A%=1:B%=-1:AB%=281:BA%=-281
50 CRT=32768:WD=40
100 GOSUB300
105 FORJ=PEEK(42)+256*PEEK(43)TOPEEK(44)+256*PEEK(45)STEP7:PRINT
110 FORI=0TO36STEP6
115 PRINTTAB(I)STR$(PEEK(J+INT(I/6)));
120 NEXT:IFPEEK(197)=131THENM=1:GOSUB500
130 PRINT:NEXT:M=2:PRINT"END OF TABLE":GOSUB500
200 FORX=1024TOPEEK(44)+256*PEEK(45)STEP2
205 PRINTTAB(0)PEEK(X);TAB(7)FNB(X);TAB(14)FNA(X);TAB(22)PEEK(X+1);TAB(29)FNB(X
+1);TAB(36)FNA(X+1)
210 IFPEEK(197)=131THENGOSUB500
215 NEXT:PRINT"END OF TABLE":END
300 L=L+1
301 PRINT"BASIC VARIABLE TABLE":PRINT"SCREEN #";L:PRINT
305 PRINT"6543210":RETURN
400 P=P+1:IFX>256*PEEK(43)THEN402
401 PRINT"PROGRAM AND VARIABLE STORAGE":PRINT"SCREEN #";P:GOTO410
402 PRINT"PROGRAM AND VARIABLE STORAGE":PRINT"SCREEN #";P
410 PRINT"DATALOC.PAGEDATALOC.PAGE":RETURN
500 PRINT:PRINT"HIT ANY KEY TO CONTINUE"
501 GOSUB59000
505 GETA$:IFA$=""THEN505
510 ON M GOSUB300,400
515 RETURN
58998 REM SCREEN PRINT ROUTINE COURTESY CURSOR MAGAZINE #24 : THE CODE WORKS
58999 REM GOLATA CA
59000 REM PRINT THE SCREEN
59005 OPEN4,4:ZC=WD:ZR=0
59006 ZS=46:GOSUB59150:PRINT#4
59007 ZS=18:GOSUB59150
59010 FORZT=CRTTOCRT+25*WD:Z=PEEK(ZT):ZL=Z
59020 IFZC=WDTHENZC=0:PRINT#4:IFPEEK(59468)=14THENPRINT#4,"";
59030 POKEZT,(ZL+128)AND255
59040 IFZ>127ANDZR=0THENPRINT#4,"";:ZR=1
59045 IFZ<128ANDZR=1THENPRINT#4,"";:ZR=0
59050 Z=ZAND127:ZS=0:IFZ>63THENZS=128:Z=Z-64
59060 IFZ<32THENZ=Z+64
59080 PRINT#4,CHR$(Z+ZS);
59090 IFZ=34THENPRINT#4,CHR$(141);TAB(ZC+1);
59120 ZC=ZC+1:POKEZT,ZL
59130 NEXTZT:PRINT#4:ZS=45:GOSUB59150:PRINT#4:CLOSE4
59140 ZS=24:GOSUB59150:RETURN
59150 OPEN6,4,6:PRINT#6,CHR$(ZS):CLOSE6:RETURN
```

Lines 58998–59150 of Listing 1b reprinted with permission of *Cursor Magazine*, Box 550, Goleta, CA 93116.

Listing 1b. Listing 1a with screen print routine added.

HI-RES CROSSWORDS FOR APPLE II



THE CROSSWORD MACHINE™

PLAY THE PUZZLES . . . EASY, HARD & TOPICAL . . . CREATE & PLAY YOUR OWN

MAIL ORDER SPECIAL $24.95 DISK

PLUS $2.00 POSTAGE AND HANDLING

DEALER INQUIRIES: L & S COMPUTERWARE, 1589 FRASER DR., SUNNYVALE, CA 94087

SYSTEM REQUIREMENTS: 48K, ROM APPLE SOFT

APPLE II IS A TRADEMARK OF APPLE COMPUTER INC.

NEED YOUR ORDER SHIPPED NOW?

THEN CALL TOLL FREE

(800) 227-1617 EXT. 481

master charge
THE INTERBANK CARD

IN CALIFORNIA
(800) 772-3545
EXT. 481

VISA

L&S COMPUTERWARE

P.O. BOX 70728, SUNNYVALE, CA 94086

SEND THE CROSSWORD MACHINE™

ENCLOSED $24.95 AND $2.00 POSTAGE AND HANDLING ☐ CHECK ☐ MONEYORDER

NAME ______________________

ADDRESS ______________________

CITY ____________ STATE ______ ZIP ______

program is changed, as in line 0, I found that the first run of the program would produce unusual output which could be corrected by running the program again. The directed run corrects this problem.

To observe this distortion, add a line 10 to Listing 2 so that one of the values from lines 106 to 110 is used to alter the program size after it is running (e.g., make line 10 POKE 124, 161 : POKE 125,14). Then run the program. Now run the program again, and the funny output to the screen is replaced by the correct output.

Listing 2 is an extension of the idea of making a system of programs size-compatible. Here, a menu is adjusted to the size of the program selected. This is an automatic directory system for your cassettes. The data is changed to the programs that are on the multi-program file which you create.

As written, Listing 2 has to be run twice to be successful. Subroutine 105 plays "hide and seek" with the variable table. The first pass through the routine sets the memory size of the menu program to that of the program you want to run. When the PET looks into this location, it doesn't find the variable table there and returns with an error:

ILLEGAL QUANTITY ERROR IN 90.

When this program is run again, the variable table has been relocated to the program you selected, and the PET loads and executes this program.

Modifications

A method of correcting this problem, using the dynamic keyboard and PET shorthand, might be better than having to run the program twice. The dynamic keyboard is discussed in the "Cover Up" article. There are several unused bytes in the PET, from the end of the second cassette buffer to the start of the BASIC text area of memory.

One of these is used to store the value of the program selected, and another is used as a flag bit to indicate if you are in pass one or pass two of the program. The dynamic keyboard will be set up with the text Ru5 (shorthand for RUN5).

At the end of the first pass, the error condition will stop the program. But the statement RUN 5 in the dynamic keyboard will restart the program, and now, instead of being input, the value we selected as J will be entered from the storage area, and the selected program from the menu will be loaded and will execute.

The changes needed in Listing 2 to do this are:

change line 0 to line 5
add line 0: 0 POKE 1018,5
change line 80 to: 80 IF PEEK 1018=5 THEN INPUT(3dnENTER(rev)NUMBER(off) OF CHOICE";J:POKE 1019,J
add line 85: 85 IF PEEK 1018=10 THEN J=PEEK(1019)
change the reference in line 90 from Gosub 105 to GOSUB 103
add line 103: 103 IF PEEK (1018)=5 THEN POKE 1018,10: GOSUB 200
add the dynamic keyboard initialization subroutine by adding line 200: 200 POKE525,4: POKE527,82:POKE528,213:POKE529,53: POKE530,13:RETURN
(The keyboard buffer on the new PETs is moved from 527–536 to 623–632. This line would become 200 POKE 158,4:POKE 623,82: POKE 624,213:POKE 625,53:POKE 626,13:RETURN.)

These modifications will cause the directory program to run without the operator restarting it.

```
0 PRINT"▛"
20 DATA5,22,11,COVER,3K,TIC-TAC-TOE,6K,DISCO,6K,ADFLAG,1K,RDS,6K,EXIT MENU
30 READP,M,I:DIMA$(I)
40 PRINT"NO.";TAB(P)"PROGRAM NAME";TAB(M)"MEMORY SIZE":PRINT"▚"
50 FORJ=1TOI:READA$(J):NEXT
60 FORJ=1TOI-1STEP2:PRINT(J+1)/2;TAB(P)A$(J);TAB(M)A$(J+1):NEXT
70 PRINT(J+1)/2;TAB(P)"▚";A$(J);"■"
80 INPUT"▚▚▚ENTER ▚NUMBER■ OF CHOICE";J
90 IFLEFT$(A$(J*2-1),4)<>"EXIT"THENGOSUB105:LOADA$(2*J-1)
100 END
105 ONJGOTO106,107,108,109,110
106 POKE124,161:POKE125,14:RETURN
107 POKE124,152:POKE125,21:RETURN
108 POKE124,160:POKE125,24:RETURN
109 POKE124,101:POKE125,07:RETURN
110 POKE124,154:POKE125,24:RETURN
```

Adjustment for 3.0 ROMs and 4.0 BASIC.

```
106  POKE 42, 161: POKE 43,14 : RETURN
107  POKE 42, 152: POKE 43,21 : RETURN
108  POKE 42, 160: POKE 43,24 : RETURN
109  POKE 42, 101: POKE 43,07 : RETURN
110  POKE 42, 154: POKE 43,24 : RETURN
```

Listing 2. Executive directory.

Another change you might want to add to the cassette directory program while setting it up for your own cassette programs, is a fast-forward subroutine. This will reduce the waiting time when you select a program which is well down the line in the cassette. The "Cover Up" article contains such a fast-forward routine.

William Pytlik's "Case of the Missing Tape Counter" (June 1979 *Micro*) presents in detail a fast-forward timed routine for improving the speed of finding cassette programs.

In conclusion, the command LOAD is the key to increasing the length of your application programs. The ability to pass parameters correctly from one program to another, and the ability to correct the size-incompatibility problem, should enable you to use the memory contained in your system to greater advantage. ■

References

PET User Notes, Vol. 1, No. 7, "New Discovery?"

PET User Manual, Commodore Business Machines.

Len Lindsay "Teach Your Old PET New Tricks" *Kilobaud Microcomputing*, July 1979.

Arnie Lee, *PET Machine Language Guide*, 1979, Abacus Software.

TRS-80 Newsletter, June 1979: Radio Shack, Fort Worth, TX.

C. Ascolillo, N. Schwartz, "Cover Up," *Kilobaud Microcomputing*, August 1979.

Gary Ratliff, "PET Shorthand Compleat," *Kilobaud Microcomputing*, March 1981.

William Pytlik "Case of the Missing Tape Counter" *Micro*, June 1979.

Mark Zimmerman, "Snooping With Your PET," *People's Computers*, July–August 1978.

Cursor, #24, Feb. 1981.

✓See List of Advertisers on page 210

CALENDAR

Computer Summer Camp

This summer youngsters can again sign up for an overnight camp in Moodus, CT, where the main activity will be computers. This unique recreational and educational experience is directed by Dr. Michael Zabinski, Professor at Fairfield University.

The 1981 National Computer Camp will feature two one-week sessions: July 19–24 and July 26–31. The campers, ages 10–17, will have small group instruction and hands-on experience.

For information, contact Michael Zabinski at 203-795-9069, or write to Computer Camp, Grand View Lodge, Box 22, Moodus, CT 06469.

London Micro Show

The 1981 Microcomputer Show will be held July 29–31, 1981, at the Wembley Conference Centre, London, England. A conference program running concurrently with the exhibition will highlight micro applications in business, production and education. Proposed topics for conference sessions include hardware availability, software packages, and development, automatic test equipment, robotics and process control.

The 1981 Microcomputer Show is organized by Online Conferences, also the organizers of Viewdata 80, the first world conference on viewdata, videotext and teletext.

Potential exhibitors should contact Jeff Wolf at 800-227-3477; in Canada and California, call 415-474-3000 or write TMAC, 680 Beach St., Suite 428, San Francisco, CA 94109.

Atlanta Small Computer Show

The Atlanta Small Computer Show will be held June 6–9, 1981, at the Atlanta Hilton. Exhibitors will include producers of small computers, peripheral equipment, supplies and services.

For information, write The Atlanta Small Computer Show, 4060 Janice Drive, Suite C-1, East Point, GA 30344, or call 404-767-9798.

Microcomputers for Managers

Exhibits and seminars to assist managers in the selection and use of microcomputers will be held in Salt Lake City, UT, on May 27–28 and in Denver, CO, on July 22–23. For further information, contact International Seminars, Inc., PO Box 7029, University Station, Provo, UT 84602 (801-375-7379).

MICRO QUIZ

(from page 30)

Answer: (1,1,1,0), (1,0,1,1), (1,0,0,1) and (1,0,0,0) A must be true. The following truth table yields the values for B, C and D.

B	C	D	$\overline{C\overline{D}}$	$(\overline{C\overline{D}}) \oplus B$
0	0	0	1	1
0	0	1	1	1
0	1	0	0	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

Applefest Comes to Boston

Applefest '81, devoted exclusively to the Apple computers, will take place Saturday and Sunday, June 6 and 7, 1981, at the Plaza Castle, Arlington St., Boston, MA. Over 100 exhibits of Apple computers, software, peripherals, accessories, applications and services will be featured. Free seminars will run the two days of the show. Admission is $3 per day.

Applefest '81 is sponsored by Apple/Boston, the Boston Computer Society's Apple user group. For information or tickets, contact: The Boston Computer Society, Three Center Plaza, Boston, MA 02108, 617-367-8080.

Los Angeles Shows

The Business and Personal Computer Sales Expo and the Los Angeles Business Show will be held at the Los Angeles Convention Center from July 9–11. For further information, contact Produx 2000, Inc., Box 2000, Bala Cynwyd, PA 19004.

CLUB NOTES

Tulsa Computer Society

The Tulsa Computer Society meets on the last Tuesday of each month at the Tulsa Vocational-Technical School (3420 S. Memorial Drive) at 7:30 PM. Contact the society at PO Box 1133, Tulsa, OK 74101.

Tallahassee Computerists

The Tallahassee Amateur Computer Society (TACS) meets on the second Thursday of the month to explore the uses of small computer systems for educational, business and personal applications. Write to the Tallahassee Amateur Computer Society, PO Box 6717, Tallahassee, FL 32301, for further information.

Permian Basin Club

The Permian Basin Amateur Computer Group meets on the second Tuesday of each month in Midland, TX, in the Midland College Student Center at 7:30 PM and in Odessa, TX, on the second Saturday of each month at 1 PM in the Odessa College Electronics Technology Building. This group was recently organized to exchange information on small computers and to provide technical assistance on computer projects. Contact the Permian Basin Amateur Computer Group, c/o John Rabenaldt, Ector County School District, Box 3912, Odessa, TX 79760.

Attention, Robot Enthusiasts

David Smith is interested in forming a club in the New York City area for anyone interested in building or working with robots. If you are interested in joining such a club, contact David at 45405 Kennedy Blvd., North Bergen, NJ 07047 (201-856-4890).

Toronto Computer Enthusiasts

The Toronto Region Association of Computer Enthusiasts (TRACE) holds two meetings each month: at the Ontario Science Center, 770 Don Mills Road, Don Mills, at 2 PM on the second Sunday of each month; and on the fourth Friday of each month in Room J206, Humber College, Rexdale, at 7:30 PM. For further information, contact the Toronto Region Association of Computer Enthusiasts, PO Box 6922, Station A, Toronto, Ontario, M5M 1X6, Canada.

✓*See List of Advertisers on page 210*

CLASSIFIEDS

Classified advertisements are intended for use by persons desiring to buy, sell or trade used computer equipment. No commercial ads are accepted.

Two sizes of ads are available. The $5 box allows up to 5 lines of about 35 characters per line, including spaces and punctuation. The $10 box allows up to 10 lines. Minimize use of capital letters to save space. No special layouts allowed. Payment is required in advance with ad copy. We cannot bill or accept credit.

Advertising text and payment must reach us 60 days in advance of publication (i.e., copy for March issue, mailed in February, must be here by Jan. 1). The publisher reserves the right to refuse questionable or inapplicable advertisements. Mail copy with payment to: **Classifieds, Kilobaud Microcomputing, Peterborough NH 03458.** Do not include any other material with your ad as it may be delayed.

For Sale: Synertek Keyboard Terminal Module (KTM 2/40), upper/lowercase, graphics, reverse video, RS-232, 110 to 9600 baud; $175. John Maslowski, 108 Meadowcrest Dr., Nanticoke, PA 18634. 717-735-2341.

9-inch video monitors for sale. No case, in factory carton, 100% prime tested but never used, green phosphor CRT, requires composite video and 13–16 V dc inputs, schematic incl. I can ship COD, but I will pay shipping if you send a money order for $60 to: Chris Kalmbach, 422 S. Florida Ave., Joplin, MO 64801.

Heath H-8 computer and H9 video terminal. Serial and cassette I/O card, 32K static RAM, manuals and software. Like new, perfect working condition. $725. 215-672-6312.

Wanted: Bare printed circuit board for D. C. Hayes 80-103A modem. Contact D. Reinebeck, 899 Fairfield Rd., North Vancouver, B. C., V7H 2J4, Canada. 604-929-5759, Telex 04-352544.

Hardly-used Hayes Micromodem for Apple. Like new! Only $285. Still has unused warranty card! Also Centronics 737, like new, only $700. For $85 more you can have the Apple interface & cable. Howard Rothman, 218 Huntington Road, Bridgeport, CT 06608. 203-333-6436.

Quest Super Elf 1802 system in cab., 4K exp. bd., monitor & Tiny BASIC ROM, pwr supply, rf mod, ASCII kbd, manuals, mags, tapes. $300. R. Moffie, 20121 Leadwell St., #3, Canoga Park, CA 91306.

For Sale: SWTP 6800-based C/W CT64, AC-30, recorder, printer, BASIC assembler, WordPro, parallel and serial ports. Price $1200. Call 416-457-5784 or write, 74 Kingsmere Cres., Brampton, Ontario, L6X 1Z4, Canada.

For Sale: SS-50 16K static RAM board w/16K 2114; $100. Centronics P-1 printer; $150. Motorola M-4003-340 monitor; $175. Microswitch 113-key kybd; $125. Call John 408-262-3101.

HELP! Need paper tape, hard copy hex dump, or Tarbell cassette tape of Xitan Disc BASIC, Version 1.06 (Computer Design Labs won't help!). Will trade for programs, cash, or whatever. R. Schubert, 7016 N. Roremead Blvd., Apt. 28, San Gabriel, CA 91775.

VisiCalc user's note. New user's group formed for you and those interested in business applications. TRS-80, Apple. Quarterly newsletter, $7.50. Micro-Calc Business User's Group, PO Box 12039, Salem, OR 97309.

TRS-80 Model I, 32K, E/I, disk, MicroDOS, data separator, also LNW expansion 80% complete. $1150. 10% down, bal. UPS COD. D. Bassetti, 730 Ave. Z, Brooklyn, NY 11223.

For Sale: Assembled, operational; Heath H-8 computer w/H8-4 four port serial I/O board; 64K Heath static RAM; Pam-Go ROM; H-17 with 2 disk drives; H-9 terminal; HDOS; MBASIC; plus many HUG disks. 14 months old in excellent condition. $1895 + UPS shipping charges. Certified check or money order only. Contact Paul P. Bittner, R.D. #2, Box 775, Slatington, PA 18080, or call 215-767-5357 8 AM to 4 PM.

Sell: Mod 33 ASR Teletype. Extra ribbons & paper, manuals & schematics. $450. Will deliver within 200 mi. of Alamogordo, NM. David Marshall, 1803 Scenic Dr., Alamogordo, NM 88310. 505-437-6374.

Digital Group computer. Factory wired, less than 5 hrs. running time. Z-80A processor, 32K static RAM, keyboard, dual Phideck (early board). Complete documentation, showroom condition. $700 obo, F.O.B. Los Angeles. J. Zubrecky, 328-A Russell Ave., Monterey Park, CA 91754. 213-571-1554.

For Sale: TRS-80 48K, EI, one disk drive, M80, Flesher TU-170. $1100, pickup only. P. Foster, 1103 Serenade, Richardson, TX 75081.

SWTPC 6800, 16K $200; MPA2-B, $20; MPR, $30; AC-30, $50; TC-3/CFM-3, $40; PMB-1, $70; CT-64 (no case), $100; GT-6144/MPL, $50; PPG-J, $30; all socketed; Games & Basic: SWTPC, $15; TSC, $40; Asis Singer Frieden 30 cps, $60; Shp'g Add't'l; Total Package, $600; I pay shipping; Call Jim, Sr. after 6 PM. 703-243-4748.

For Sale: Memory, Godbout Econoram II for S-100 3-8K boards. Sacrifice, $75 each. Documentation on Radio Shack expansion interface (26-2001) and Model I (26-2103); both for $10. T-Bug, $9. Call Mike 901-795-5122.

Marchant, Cogito 566PR programmable, printing, adding machine, square root, 16 digit, needs adjustment, originally cost over $2000, sell for $100 plus postage. Rockwell PPS-4/1 single board development microcomputer, never used, costs $500, sell for $395. John Boyd, 6611 Burkett St., Houston, TX 77021. 713-747-3977.

For Sale: SWTP 6802 syst/parts, B2 mainfrm, MPA2 MPU, 40K RAM, serial/parallel cards, Percom LFD400 FDC, EPROM prg, calc. card—$900; H19 term., Base 2 prt—$1300 w/syst HiPlot DMP2 plot—$800; Hewlett-Packard 85 computer, 4 software packs, 6 data cassettes, 2 paper rolls—little use in 9 months—$2900. Doug Ramers, Box 1008, Dover, DE 19901. 302-674-6386 (days).

Memory boards from bankruptcy sale. Limited quantity of 16K dynamic RAMs. Configured for 18 bits, 8K words or 9 bits, 16K bytes, on board refresh & add. to 20 bits. Schematics, interface information & 110-pin gold plated, wire-wrap connector. Functionally tested 1-3 @ $100, 3-5 @ $85. Untested and others from $40. TMI, 2000 N. Highway 67, Florissant, MO 63033. 314-837-8338.

Repossessed Diablo daisywheel printer, $1295. Contact Amy Vea. Data Terminals & Communications. Phone 408-378-1112.

DEALER DIRECTORY

El Monte, CA

Ohio Scientific specialist in the San Gabriel Valley serving greater Los Angeles. Full product line on display. Specializing in business computers. In-house service. Custom programming. Terminals. Printers. Open Mon–Sat, 9 AM–7 PM. **Computer & Video, 3380 Flair Dr., Suite 207, El Monte, CA 91731, 572-7292.**

San Jose, CA

New and used computer products—specializing in S-100 boards, printers, drives, chassis and complete systems, as well as supplies and parts —Imsai, Tandon, Diablo—5000 sq. ft. **W/W Component Supply, Inc., 1771 Junction Ave., San Jose, CA 95112.**

Sarasota, FL

Your personal and business computer store for Dynabyte, Vector, HP-85, Atari and Epson. Structured Systems and Micro-Pro software. Computer furniture and books by Osborne or Hayden. Sales, service and supplies. **Computer Crossroads, 3800 S. Tamiami Trail, Sarasota, FL 33579, 349-0200.**

Sarasota, FL

Real-Time/Industrial Software Engineering Consultants. Top-down design and structured techniques utilized. **Advanced Computer Concepts, Inc., PO Box 15710, Sarasota, FL 33579.**

Call or visit
the dealer nearest you!

Tampa, FL

Apple Computer sales and service. S-100 boards from SSM, Godbout, Thinker Toys, California Computer Systems. Computer books and magazines. **AMF Microcomputer Center, Inc., 11158 N. 30th Street, Tampa, FL 33612, 971-4072, 977-0708.**

Aurora, IL

Microcomputer systems for home or business; peripherals, software, books & magazines. Apple, Hewlett-Packard Series 80 Systems, HP Calculators, IDS, Qume, Starwriter printers. **Farnsworth Computer Center, 1891 N. Farnsworth Ave., Aurora, IL 60505, 851-3888.**

Chicago, IL

Computer Hardware/Specialists for home and business. Largest selection of computer books, magazines and copyrighted software in Illinois. Experienced factory-trained service department. Feature Apple, Alpha Microsystems and Hewlett-Packard calculators and accessories. **Data Domain of Schaumburg, 1612 E. Algonquin Rd., Schaumburg, IL 60195, 397-8700.**

Chicago, IL

Brand new lowest prices, never undersold, postpaid in USA—Teletype 43 keyboard printers, Okidata & Integral Data printers, SS-50 bus computers, peripherals & business software. **Data Mart, 914 East Waverly Street, Arlington Heights, IL 60004, 398-8525.**

Herington, KS

Hardware support. Maintenance and service for all microcomputers and peripherals. Kits assembled or debugged. Radio Shack (mods OK) repaired. Quality work, fast turnaround and reasonable cost. **Prairie Micro Clinic, Box 325, Herington, KS 67449, 258-2179.**

Garden City, MI

Books, magazines, hardware and software for Apple, North Star, TRS-80 and PET. **Computer Center, 28251 Ford Rd., Garden City, MI 48135, 425-2470.**

Hannibal, MO

Ohio Scientific products, modifications, service, software. 8" disk for C1P, C4P. Process control specialist. **E&I Technical Service, 5300 Paris Gravel Road, Hannibal, MO 63401, 248-0084.**

Chautauqua, NY

Retail book store featuring the Disassembled Handbook for TRS-80 Volumes 1, 2, 3. English, German & French language editions. 9 AM–5 PM weekdays. Come and visit us. **Richcraft Computer Book Store, 1 Wahmeda Ave., Chautauqua, NY 14722, 753-2654.**

Long Island, NY

Software-hardware application engineering. System startups & tutorial sessions. Program customizing, translation & development. Data communication networks. World-wide service available. Local experts needed. **Charles Stewart, 442 Lockwood Dr., No. Shirley, NY 11967, 924-5164.**

Portland, OR

Ohio Scientific specialists for business and personal computers. Local service. Terminals, printers, custom programming. Full OSI product line on display! 10 AM to 6 PM M–F. **Fial Computer, 11266 SE 21st Ave., Milwaukie, OR 97222, 654-9574.**

Spokane, WA

SS-50 Users: Expand present system to maximum or build from ground up. We provide PCBs for motherboards, interfaces, etc. Write for specs and information. **Quality Research Company, PO Box 7207, Spokane, WA 99207.**

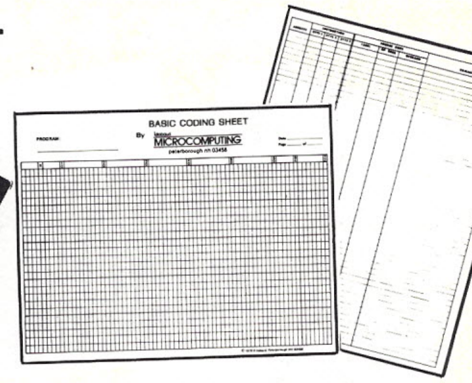
Milwaukee, WI

Specializing in the CBM-PET business, personal, educational, industrial, telecomputing systems. Consulting, modems, printers, books, accessories, magazines, supplies, peripherals. Factory authorized service. Convenient freeway access. **PETTED Micro Systems, 4265 W. Loomis Rd., Milwaukee, WI 53221, 282-4181.**

Dealers: Listings are **$15** per month in prepaid quarterly payments, or one yearly payment of **$150**, also prepaid. Ads include 25 words describing your products and services plus your company name, address and phone. (No area codes or merchandise prices, please.) Call Marcia at 603-924-7138 or write *Kilobaud* MICROCOMPUTING, Ad Department, Peterborough NH 03458.

kb microcomputing book nook

—PROGRAMMING & COOKBOOKS—



—Z-80—

● **INSIDE LEVEL II**—BK1183—For machine language programmers! This is a comprehensive reference guide to the Level II ROMs, allowing easy utilization of the sophisticated routines they contain. It concisely explains set-ups, calling sequences, variable passage and I/O routines. Part II presents an entirely new composite program structure which unloads under the SYSTEM command and executes in both BASIC and machine code with the speed and efficiency of a compiler. Special consideration is given to disk systems. $15.95.*

● **PROGRAMMING THE Z-80**—BK1122—by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive, step-by-step course. This book is both an educational text *and* a self-contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. $14.95.*

● **Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1045—by Nat Wadsworth. Scelbi's newest cookbook! This book contains a complete description of the powerful Z-80 instruction set and a wide variety of programming information. Use the author's ingredients including routines, subroutines and short programs, choose a time-tested recipe and start cooking! $16.99.*

● **Z-80 ASSEMBLY LANGUAGE PROGRAMMING**—BK1177—by Lance A. Leventhal. This book thoroughly covers the Z-80 instruction set, abounding in simple programming examples which illustrate software development concepts and actual assembly language usage. Features include Z-80 I/O devices and interfacing methods, assembler conventions, and comparisons with 8080A/8085 instruction sets and interrupt structure. $16.99.*

TEST EQUIPMENT —LIBRARY—

● **VOL. I COMPONENT TESTERS**—LB7359—. . . how to build transistor testers (8), diode testers (3), IC testers (3), voltmeters and VTVMs (9), ohmmeters (8 different kinds), inductance (3), capacity (9), Q measurement, crystal checking (6), temperature (2), aural meters for the blind (3) and all sorts of miscellaneous data on meters . . . using them, making them more versatile, making standards. Invaluable book. $4.95.*

● **VOL. II AUDIO FREQUENCY TESTERS**—LB7360— . . . jam packed with all kinds of audio frequency test equipment. If you're into SSB, RTTY, SSTV, etc., this book is a must for you . . . a good book for hi-fi addicts and experimenters, too! $4.95.*

● **VOL. III RADIO FREQUENCY TESTERS**—LB7361—Radio frequency waves, the common denominator of Amateur Radio. Such items as SWR, antenna impedance, line impedance, rf output and field strength; detailed instructions on testing these items includes sections on signal generators, crystal calibrators, grid dip oscillators, noise generators, dummy loads and much more. $4.95.*

● **VOL. IV IC TEST EQUIPMENT**—LB7362—Become a troubleshooting wizard! In this fourth volume of the 73 TEST EQUIPMENT LIBRARY are 42 home construction projects for building test equipment to work with your ham station and in servicing digital equipment. Plus a cumulative index for all four volumes for the 73 TEST EQUIPMENT LIBRARY. $4.95.*

A NEW PROGRAMMING —SERIES—

The Microprocessor Software Engineering Series by John Zarrella provides common sense descriptions of advanced computer system topics for engineers, programmers and development managers. Each volume is a self-contained review of a software engineering topic, explaining fundamental concepts in easy-to-understand language and describing sophisticated software tools and techniques. Detailed glossary of technical jargon is included in each volume. This series will help you find the solutions to your software problems.

● **OPERATING SYSTEMS: CONCEPTS AND PRINCIPLES**—BK1193—Presents an overview of the basic operating system types, their components and capabilities. $7.95.*

● **WORD PROCESSING AND TEXT EDITING**—BK1194—Provides a firm basis for understanding word processing terminology and for comparing systems. $7.95.*

● **SYSTEM ARCHITECTURE**—BK1195 —Presents a detailed overview of advanced computer system design including object architecture and capability-based addressing. $9.95.*

—6502—

● **PROGRAMMING THE 6502 (Third Edition)**—BK1005—Rodnay Zaks has designed a self-contained text to learn programming, using the 6502. It can be used by a person who has never programmed before, and should be of value to anyone using the 6502. The many exercises will allow you to test yourself and practice the concepts presented. $12.95.*

● **6502 APPLICATIONS BOOK**—BK1006—Rodnay Zaks presents practical-application techniques for the 6502 microprocessor, assuming an elementary knowledge of microprocessor programming. You will build and design your own domestic-use systems and peripherals. Self-test exercises included. $12.95.*

● **6502 ASSEMBLY LANGUAGE PROGRAMMING**—BK1176—by Lance A. Leventhal. This book provides comprehensive coverage of the 6502 microprocessor assembly language. Leventhal covers over 80 programming examples from simple memory load loops to complete design projects. Features include 6502 assembler conventions, input/output devices and interfacing methods, and programming the 6502 interrupt system. $16.99.*

● **6502 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1055—by Robert Findley. This book introduces the BASIC language programmer into the realm of machine-language programming. The description of the 6502 structure and instruction set, various routines, subroutines and programs are the ingredients in this cookbook. "Recipes" are included to help you put together exactly the programs to suit your taste. $12.95.*

● ***MICROCOMPUTING* CODING SHEETS** *Microcomputing*'s dozen or so programmers wouldn't try to work without these handy scratch pads, which help prevent the little errors that can cost hours and hours of programming time. Available for programming is Assembly/Machine Language (PD1001), which has columns for address, instruction (3 bytes), source code (label, op code, operand) and comments; and for BASIC (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39.*

—8080/8080A—6800—

● **8080A/8085 ASSEMBLY LANGUAGE PROGRAMMING**—by Lance Leventhal—BK1004—Assembly language programming for the 8080A/8085 is explained with a description of the functions of assemblers and assembly instructions, and a discussion of basic software development concepts. Many fully debugged, practical programs are included as is a special section on structured programming. $15.99.*

● **8080 PROGRAMMING FOR LOGIC DESIGN**—BK1078—Ideal reference for an in-depth understanding of the 8080 processor. Application-oriented and the 8080 is discussed in light of replacing conventional, hard-wired logic. Practical design considerations are provided for the implementation of an 8080-based control system. $9.50.*

● **8080 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1102—If you have been spending too much time developing simple routines for your 8080, try this new book by Scelbi Computing and Robert Findley. Describes sorting, searching, and many other routines for the 8080 user. $12.95.*

● **6800 PROGRAMMING FOR LOGIC DESIGN**—BK1077—Oriented toward the industrial user, this book describes the process by which conventional logic can be replaced by a 6800 microprocessor. Provides practical information that allows an experimenter to design a complete micro control system from the "ground up." $9.50.*

● **6800 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1075—Like its culinary cousin, *The 8080 Gourmet Guide*, this book by Scelbi Computing and Robert Findley describes sorting, searching and other routines—this time for the 6800 user. $12.95.*

—COOKBOOKS—

● **CMOS COOKBOOK**—BK1011—by Don Lancaster. Details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. Required reading for every serious digital experimenter! $10.50.*

● **TVT COOKBOOK**—BK1064—by Don Lancaster. Describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. $9.95.*

● **TTL COOKBOOK**—BK1063—by Don Lancaster. Explains what TTL is, how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter and a digital tachometer. $9.50.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information.

No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4–6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook

—BASIC & PASCAL—

new

new

● **50 BASIC EXERCISES**—BK1192—by J. P. Lamoitier. This book is structured around the idea that the best way to learn a language is through actual practice. It contains 50 completely explained exercises: statement and analysis of the problem, flowcharts, programs and actual runs. Program subjects include mathematics, business, games, and operations research, and are presented in varying levels of difficulty. This format enables anyone to learn BASIC rapidly, checking their progress at each step. $12.95*

● **THE BASIC HANDBOOK**—BK1174—by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. $14.95.*

● **LEARNING LEVEL II**—BK1175—by David Lien. Written especially for the TRS-80, this book concentrates on Level II BASIC, exploring every important BASIC language capability. Updates are included for those who have studied the Level I User's Manual. Sections include: how to use the Editor, dual cassette operation, printers and peripheral devices, and the conversion of Level I programs to Level II. $15.95.*

● **BASIC BASIC (2ND EDITION)**—BK1026—by James S. Coan. This is a textbook which incorporates the learning of computer programming using the BASIC language with the teaching of mathematics. Over 100 sample programs illustrate the techniques of the BASIC language and every section is followed by practical problems. This second edition covers character string handling and the use of data files. $10.50.*

● **ADVANCED BASIC**—BK1000—Applications, including strings and files, coordinate geometry, area, sequences and series, simulation, graphing and games. $10.75*.

● **SIXTY CHALLENGING PROBLEMS WITH BASIC SOLUTIONS (2nd Edition)**—BK1073—by Donald Spencer, provides the serious student of BASIC programming with interesting problems and solutions. No knowledge of math above algebra required. Includes a number of game programs, as well as programs for financial interest, conversions and numeric manipulations. $6.95.*

● **PROGRAMMING IN PASCAL**—BK1140—by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming, which has been defined by Edsger Dijkstra and C.A.R. Hoare. As such, it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. It does not assume knowledge of any other programming language; it is therefore suitable for an introductory course. $12.95.*

● **PASCAL**—BK1188—by Paul M. Chirlian. Professor Chirlian's textbook combines a simple approach to the PASCAL language with comprehensive coverage on how a computer works, how to use a flowchart, working from a terminal as well as batch operation and debugging. Special attention is paid to idiosyncrasies of the language and syntax flowcharts abound for the convenience of the experienced programmer. Well indexed. $12.95*

● **INTRODUCTION TO PASCAL**—BK1189—by Rodnay Zaks. A step-by-step introduction for anyone wanting to learn the language quickly and completely. Each concept is explained simply and in a logical order. All features of the language are presented in a clear, easy-to-understand format with exercises to test the reader at the end of each chapter. It describes both standard PASCAL and UCSD PASCAL, the most widely used dialect for small computers. No computer or programming experience is necessary. $14.95.*

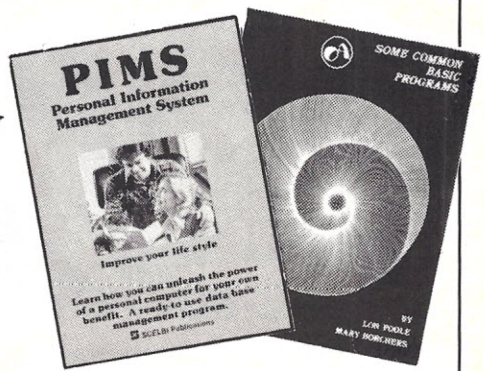
—GAMES—

● **40 COMPUTER GAMES**—BK7381—Forty games in all in nine different categories. Games for large and small systems, and even a section on calculator games. Many versions of BASIC used and a wide variety of systems represented. A must for the serious computer gamesman. $7.95*

● **BASIC COMPUTER GAMES**—BK1074—Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

● **MORE BASIC COMPUTER GAMES**—BK1182—edited by David H. Ahl. More fun in BASIC! 84 new games from the people who brought you *BASIC Computer Games.* Includes such favorites as Minotaur (battle the mythical beast) and Eliza (unload your troubles on the doctor at bargain rates). Complete with game description, listing and sample run. $7.50.*

—BUSINESS—

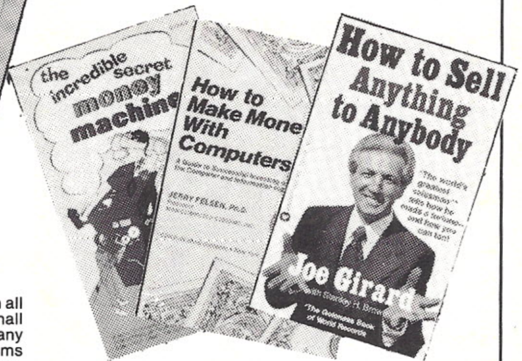


● **PAYROLL WITH COST ACCOUNTING—IN BASIC**—BK1001—by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flowcharts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor of distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941 (in **CBASIC**). $20.00.*

● **SOME COMMON BASIC PROGRAMS**—BK1053—published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. $14.99 paperback.

● **PIMS: PERSONAL INFORMATION MANAGEMENT SYSTEM**—BK1009—Learn how to unleash the power of a personal computer for your own benefit in this ready-to-use data-base management program. $11.95.*

—MONEYMAKING—



● **HOW TO MAKE MONEY WITH COMPUTERS**—BK1003—In 10 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. $15.00.*

● **HOW TO SELL ANYTHING TO ANYBODY**—BK7306—According to *The Guinness Book of World Records*, the author, Joe Girard, is "the world's greatest salesman." This book reveals how he made a fortune—and how you can, too. $2.25.*

● **THE INCREDIBLE SECRET MONEY MACHINE**—BK1178—by Don Lancaster. A different kind of "cookbook" from Don Lancaster. Want to slash taxes? Get free vacations? Win at investments? Make money from something that you *like* to do? You'll find this book essential to give you the key insider details of what is really involved in starting up your own money machine. $5.95.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information.

No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook

—INTRODUCTORY—

new

● **UNDERSTANDING AND PROGRAMMING MICROCOMPUTERS**—BK7382—A valuable addition to your computing library. This two-part text includes the best articles that have appeared in 73 and Kilobaud Microcomputing magazines on the hardware and software aspects of microcomputing. Well-known authors and well-structured text helps the reader get involved. $10.95*

● **SOME OF THE BEST FROM KILOBAUD MICROCOMPUTING**—BK7311—A collection of the best articles that have appeared in Kilobaud MICROCOMPUTING. Included is material on the TRS-80 and PET systems, CP/M, the 8080/8085/Z-80 chips, the ASR-33 terminal. Data base management, word processing, text editors and file structures are covered too. Programming techniques and hard-core hardware construction projects for modems, high speed cassette interfaces and TVTs are also included in this large format, 200 plus page edition. $10.95.*

● **YOUR FIRST COMPUTER**—BK1191—by Rodnay Zaks. Whether you are using a computer, thinking about using one or considering purchasing one, this book is indispensable. It explains what a computer system is, what it can do, how it works and how to select various components and peripheral units. It is written in everyday language and contains invaluable information for the novice and the experienced programmer. (The first edition of this book was published under the title "An Introduction to Personal and Business Computing".) $7.95*

● **HOBBY COMPUTERS ARE HERE!**—BK7322—If you want to come up to speed on how computers work... hardware and software... this is an excellent book. It starts with fundamentals and explains the circuits, and the basics of programming, along with a couple of TVT construction projects, ASCII-Baudot, etc. This book has the highest recommendations as a teaching aid. $4.95.*

● **THE NEW HOBBY COMPUTERS**—BK7340—This book takes it from where "HOBBY COMPUTERS ARE HERE!" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards... and much, much more! Don't miss this tremendous value! Only $4.95.*

INTRODUCTION TO — MICROCOMPUTERS — (VOL. 0–III)

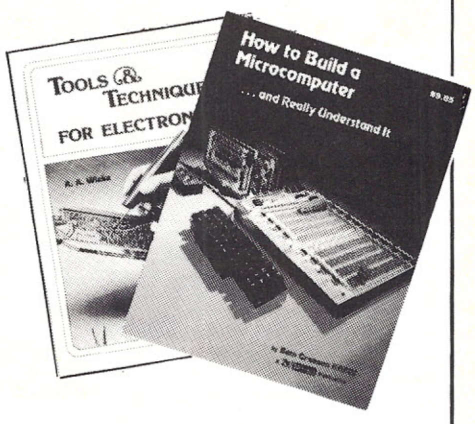
● **AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0**—BK1130—The Beginner's Book—Written for readers who know nothing about computers—for those who have an interest in how to use computers—and for everyone else who must live with computers and should know a little about them. The first in a series of 4 volumes, this book will explain how computers work and what they can do. Computers have become an integral part of life and society. During any given day you are affected by computers, so start learning more about them with Volume 0. $7.95.*

● **VOL. I**—BK1030—**2nd Edition completely revised.** Dedicated to the basic concepts of microcomputers and hardware theory. The purpose of Volume I is to give you a thorough understanding of what microcomputers are. From basic concepts (which are covered in detail), Volume I builds the necessary components of a microcomputer system. This book highlights the difference between minicomputers and microcomputers. $12.99.*

● **VOL. II**—BK1040 (with binder)—Contains descriptions of individual microprocessors and support devices used only with the parent microprocessor. Volume II describes all available chips. $31.99*

● **VOL. III**—BK1133 (with binder)—Contains descriptions of all support devices that can be used with any microprocessor. $21.99*

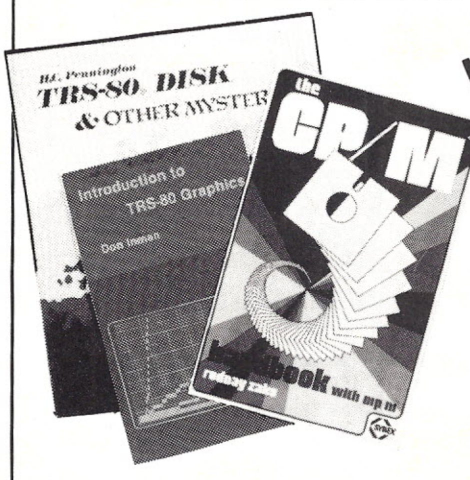
FOR THE ELECTRONICS — HOBBYIST —



● **TOOLS & TECHNIQUES FOR ELECTRONICS**—BK7348—by A. A. Wicks is an easy-to-understand book written for the beginning kit builder as well as the experienced hobbyist. It has numerous pictures and descriptions of the safe and correct ways to use basic and specialized tools for electronic projects as well as specialized metal working tools and the chemical aids which are used in repair shops. $4.95.*

● **HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT**—BK7325—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. This book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing and debugging of a complete microcomputer system. Must reading for anyone desiring a true understanding of small computer systems. $9.95.*

—SPECIAL INTERESTS—



new

● **MICROSOFT BASIC DECODED AND OTHER MYSTERIES**—BK1186—by James Farvour. From the company that brought you *TRS-80 DISK AND OTHER MYSTERIES*! Contains more than 6500 lines of comments for the disassembled Level II ROMs, six additional chapters describing every BASIC subroutine, with assembly language routines showing how to use them. Flowcharts for all major routines give the reader a real insight into how the interpreter works. $29.50.

● **THE CP/M HANDBOOK (with MP/M)**—BK1187—by Rodnay Zaks. A complete guide and reference handbook for CP/M—the industry standard in operating systems. Step-by-step instruction for everything from turning on the system and inserting the diskette to correct user discipline and remedial action for problem situations. This also includes a complete discussion of all versions of CP/M up to and including 2.2, MP/M and CDOS. $14.95*.

● **INTRODUCTION TO TRS-80 GRAPHICS**—BK1180—by Don Inman. Dissatisfied with your Level I or Level II manual's coverage of graphics capabilities? This well-structured book (suitable for classroom use) is ideal for those who want to use all the graphics capabilities built into the TRS-80. A tutorial method is used with many demonstrations. It is based on the Level I, but all material is suitable for Level II use. $8.95.*

● **TRS-80 DISK AND OTHER MYSTERIES**—BK1181—by Harvard C. Pennington. This is the definitive work on the TRS-80 disk system. It is full of detailed "How to" information with examples, samples and in-depth explanations suitable for beginners and professionals alike. The recovery of one lost file is worth the price alone. $22.50.*

● **MICROPROCESSOR INTERFACING TECHNIQUES**—BK1037—by Austin Lesea & Rodnay Zaks—will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S-100. $15.95.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information.

No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

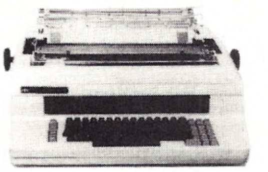
FOR TOLL FREE ORDERING CALL 1-800-258-5473

10 DAY FREE RETURN

C-ITOH STARWRITER: LETTER QUALITY PRINTING FOR UNDER $2,000!
This daisywheel printer gives high quality at a low price. 25 cps. Parallel and serial interfaces available. **Price $1895.**

NEC THE FIRST NAME IN LETTER QUALITY PRINTERS
COMPUMART offers beautiful print-quality with NEC Spinwriter terminals. Both KSR and RO versions give unsurpassed hard copy output. CALL

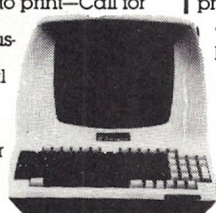
	COMPUMART PRICE	List Price
5510	**$2749.**	$3055.
5515	**$2849.**	$3135.
5520	**$3149.**	$3415.
5525	**$3249.**	$3455.
5530	**$2749.**	$3055.



SPECS

5510 Receive—Only (RO), Friction feed platen, copy control, impression control, bidirectional, 10-12 pitch, 6-8 lines per inch, autofeed, ribbon-out, cover open switch, and self-test.
5515 RO unit, specs. same as 5510s. Diablo compatible.
5520 KSR with numeric keypad. Specs same as 5510.
5525 KSR with numeric keypad. Diablo compatible, specs same as 5510.
5530 RO unit, same specs as 5510 Centronics parallel interface.

Super Selling Terminals
We have the following **Lear Siegler** Terminals in stock at prices too low to print—Call for quotes.
ADM—3A Industries favorite dumb terminal for some very smart reasons.
ADM—3A + **NEW** from Lear Siegler. CALL!
IT IS HERE!—It is the **new** Intermediate Terminal from Lear Siegler. Call for details.



Dysan Diskettes—Single side, Single density, Hard or Soft Sector **$5. ea.**
Memorex 3401's—5¼ discs **$3.25** with hub ring for Apple **$3.50.**
Memory Integrated Circuits—Call for qty. discounts when ordering over 50 units.
Motorola 4116 (200 Nanosecond, Plastic) **$4.50 ea.**
Fairchild 2114 (Standard Power, Plastic) **$4.50 ea.**

NEW FROM INTEGRAL DATA
THE IDS 560 PRINTER. All the exciting features of the 400 series plus 14½" paper capacity, 132 col. graphics printer.
Price $1695.

IDS 445. Priced lower than the 440 and equipped with a better print head.
IDS 445 w/graphics **$894.**
IDS 445 w/o graphics **$795.**
IDS 460 $1,295
The 460's features include: Correspondence quality printing, high resolution graphics capability, programmable print justification.

CENTRONICS PRINTERS
The incredible Model 737. The closest thing to letter quality print for under $1,000. 737-1 (Parallel Interface)
List Price $899. **SPECIAL $795.**
730-1 **Sale Price $695.**

Omni Printers from Texas Instruments
The 810—List $1895. **SALE! $1795.**
The 820 (Ro) Package—Includes machine-mounted paper tray and cable. A compressed print option and device forms control are standard features. **$2,155.**
The 820 (KSR) Package—Includes full ASCII Keyboard plus all of the features of the RO **$2395.**

EPSON MX-80
Now in stock!
The MX-80 dot matrix printer. Unequalled Epson reliability. Has all the features of the MX-70 plus more power and extra functions. **$549**
List price $645

HP-41C Calculators .... **$199.**
Memory Modules. For storing programs of up to 2000 lines of program memory **$29.95**

"Extra Smart" Card Reader. Records programs and data back onto blank magcards. **$199.**

The Printer. Upper and Lower case. High resolution plotting. Portable Thermal operation .... **$355.**
Application Modules **$29.95**

NEW SUPER 41-CV Systems with Quad RAMS built-in. (Maximum memory on-board leaves slots open for Application Pacs and peripherals.) The HP 41CV **$309.00**
HP 41CV Plus Card Reader. **$474.**

HP 41CV Plus Card Reader & Printer **$799.**

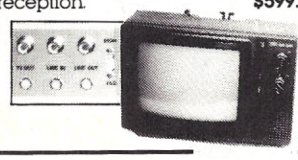
Quad RAMS for the 41-C (Equivalent to four Memory Modules all packed into one.) **$95.**

EXCLUSIVE from COMPUMART!
Special Offer. Zenith Color Video Monitor for $349!

NEW FROM SANYO—Four Great Monitors at Low COMPUMART Prices.
Sanyo's new line of CRT data display monitors are designed for the display of alphanumeric or graphic data.

9" Sanyo Monitor B/W	**$169.**
12" Sanyo Monitor B/W	**$289.**
12" Sanyo Monitor with green screen	**$299.**

THE NEC MONITOR—Composite video using BNC connectors. 8-pin connector for VCR/VTR video loop, In/Out and television reception. **$599.**



PM65 Expansion Accessories for the Aim-65
Call for specs and prices.
Forth for Rockwell AIM-65 **$195.**

ROCKWELL AIM 65
Our AIM system includes 4K AIM with BASIC interpretor assembler. Power Supply. Cassette recorder & Enclosure **$799.**
4K AIM-65 **$499.**
PL65 High Level Language **$125.**
Paper for the Aim (roll) **$2.50**
Rockwell's 4-slot Motherboard (**sale**) **$175.**
Fourth for Rockwell AIM-65 **$175.**

Visit our giant
ANN ARBOR STORE
1250 North Main Street
Ann Arbor, Michigan ✓43

SEND FOR OUR FREE CATALOG

GILTRONIX RS-232 SWITCH
If you're tired of unplugging and replugging peripherals into your computer, then you're ready for this new product from Giltronix. The RS232-X8 is flexible—you can connect three peripherals to one computer or three computers to one peripheral. It switches the eight most important RS-232 signals. Save money and aggravation with this great product. **$139**

COMPUMART stocks the complete line of **MATROX PRODUCTS.** Call for specs.

COMPUMART now offers the ENTIRE **DEC LSI-11** PRODUCT LINE. Call for prices & delivery.

NOVATION CAT™ ACOUSTIC MODEM
Answer Originate. 300 Baud, Bell 103 **$169.00**
NEW! D-CAT
Direct Connect Modem from Novation. **$199.**

SOFTWARE FROM APPLE
Apple Plot. The perfect graphic complement for Visicalc. **$70.**
Dow Jones News & Quotes **$95.**

Adventure (Uses 48K) **$35.**
DOS Tool Kit **$75.**
Apple Fortran **$200.**
Tax Planner **$120.**

FROM PERSONAL software
Visicalc **$149.**
Desk Top Plan **$99.**
NEW FROM MUSE
The Voice **$39.95**
Super Text **$150.**
Address Book **$49.95**

Miscellaneous Apple II Accessories.
Easy Writer (80 col. need a Videx) **$249.**

NEW from Apple for the Apple II:
DOS 3.3 Convert disks to 16 sector format for 23% more storage and faster access **$60.**

NEW for the Apple II
From MicroSoft
16K RAM Board **$195.**
FORTRAN **$175.**
COBOL **$750.**
Card Reader from Mountain Hardware **$1,195.**
HI-Res Dump for 460 Printer from the Computer Station **$49.95**
Trend Spotter—Statistical graphics program. Works with Visicalc. **$275.**

Exclusive—Magic Wand for the Apple— Videx. Z-80 soft card **complete system** **$925.**
(Requires 48K Apple and disk.)

HARDWARE ACCESSORIES FOR APPLE
Silentype Printer w/x face **$635**
Super Sound Generator (mono) **$159** (stereo) **$259**
Light Pen **$249**
X-10 Controller (plugs into paddle port) **$49**

Mountain Computer—Expansion accessories for your Apple
Intro!—10 System **$289**
Super Talker **$299**
The Music System **$545**
ROM plus board w/Keyboard filter **$199**
Clock Calendar **$280**
Apple Expansion Chassis **$750**
ROM Writer **$175**
16 Channel A to D Converter **$350**

From VIDEX!—Video Term.
80 Col. x 24 line, 7 x 9 Matrix plug in compatible board for the Apple II. Price **$325** without graphics EPROM. With graphics EPROM **$350.**
S.S.M. Serial & Parallel Apple Interface **$225**
ABT's Numeric Key Plan **$110**
California Microcomputer Keyboard **$195**

Save $200 on Apple Accessories with purchase of an Apple II.
Order accessories for your Apple, and we will give you $200 off the purchase price of those accessories. **SAVE $200**

APPLE III IS IN STOCK—CALL!
APPLE III Information Analyst Package—128K Apple III, Black and White Monitor 12", and information analyst software.
Price $4740.

IMPORTANT ORDERING INFORMATION
CALL: 800 343-5504, in Massachusetts (617) 491-2700, phones open from 8:30 a.m. to 7:00 p.m. Mon-Fri, 11:00 a.m. to 4:00 p.m. Sat.
PO's: Accepted from Dun & Bradstreet rated companies—shipment contingent upon receipts of signed purchase order.
SALE PRICES: Valid for month of magazine date only—all prices subject to change without notice. Our Ann Arbor retail store is open 11:00 a.m. to 7:00 p.m. Tues-Fri, 10:00 a.m. to 5:00 p.m. on Saturdays.

1 Take a PET home with you. 2 Atari isn't here to play games. 3 Incredible values from Apple II. 4 Hewlett—Packard super special.

CompuMart carries the entire Commodore line of computers, peripherals, and accessories—call our sales force today for complete prices and specs.

The CompuMart/Commodore Word Processing System
Get crisp, letter quality output and ease of operation that's unsurpassed. This system includes a Commodore 8032 32K CPU, a 4040 Dual Disk, a C-Itoh printer and X/face, and Word Pro 4 Plus (all cables included). Call for details and low prices. List $5685. **COMPUMART $4995.**

A complete system including a Dual Disk Drive, Tractor Printer, and an 80 column 32K CPU for under $4,000. No interfaces needed. Cables included. List $3,985. **CompuMart $3635**

New accessories from Commodore for Commodore

Visicalc	**$199**
Word Pro 4	**$299**
Ozz the Information Wizard	**$395**
Wordcraft 80	**$395**
Tax Preparation Planner	**$495**
Dow Jones Portfolio Mgmt System	**$149**
Personal Tax Calculator	**$69**
Assembler Development Package	**$99**

SAVE $200 on Commodore Accessories with purchase of a 32K PET. When you buy a 16K PET, apply $100 toward PET accessories. **SAVE $200!**

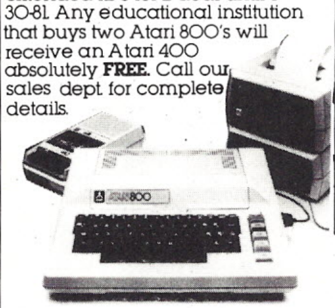
Educators Take Note!! Commodore has extended its 3 for 2 deal until 6-30-81.

Exclusive- Atari 800 Personal Computer System—Includes 800 operators Manual, 16K RAM Memory Module, 10K ROM Operating System, Power Supply, TV Switch Box **$899.**

PERIPHERALS

Atari 410 Program Recorder	**$89.95**
850 Interface w/Cable	**$249.95**
Atari 810 Disk Drive ($100 off with purchase)	**$699.95**
NEW Dual Disk double density	**$1499.95**
825 Printer (Centronics 737)	**$995.**
RS232 Interface w/Cable	**$249.95**
NEW! Light Pens	**$74.95**
NEW! Visicalc for Atari	**$199.**

Educators Take Note. Atari has extended its 3 for 2 deal until 6-30-81. Any educational institution that buys two Atari 800's will receive an Atari 400 absolutely **FREE.** Call our sales dept. for complete details.



CALL us NOW for the most complete inventory of Apple computers, peripherals, and software. You can **Save over $250** on our best selling Apple System. System includes a 48k Apple II, Apple Disk, DOS 3.3 & Controller, and a Sup R MOD RF Modulator. List $2209.

COMPUMART Sale Price $1895.

When it comes to Hewlett-Packard
Nobody does it better than COMPUMART!
Take a look at these new super specials

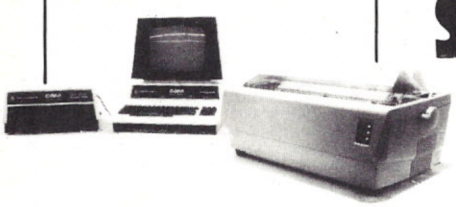
HP83 Basic System—Includes HP-83, ROM Drawer, Mass Storage ROM, 5¼" Single Master Flexible Disc Drive, HP-IB Interface Module, 2 meter HP-IB cable. **$4249.**

HP-85 System—Save $400 on HP-85 Accessories with purchase of an HP-85. Order accessories with your HP-85 and we will save you $400 off the purchase price of those accessories. **SAVE $400!**

HP-85 ACCESSORIES—We carry HP. Peripherals (Disk Drives to Graphics Plotters) Enhancements: (BASIC Training, General Statistics, Financial Decision, Math, Linear Programming **$95 ea.**); HP-85 Accessories: (Enhancement ROMs, ROM drawer, Overhead Transparency Kit); Supplies: (Plotter Pens, Tape Cartridges); Interface Modules (HP-IB Interface, HP-IB Interconnect Cables, Serial (RS-232C) Interface Module). We can get your every HP peripheral made for the HP-85.

VISICALC PLUS FOR HP-85 Everything you ever wanted from Visicalc plus plotting. **$200.**

COMPUMART'S 4 Leading Starter Sytems from THE BEST



COMPUMART has been serving the computer needs of industry since 1971.

We stock, for immediate shipment, only those products from the finest micro-computer manufacturers.

And any product, except software, can be returned within 10 days for a full refund—even if you just change your mind.

We also honor all manufacturers' warranties.

Our expert technicians will service any product we sell that cannot be better, or faster, serviced by the manufacturer's local service center.

Call us for more information on products, product configuration and service. Our phones are open Monday thru Friday, 8:30 a.m. to 7:00 p.m. and Saturday 11:00 a.m. to 4:00 p.m.

We have a staff of highly knowledgeable sales people waiting to hear from you, and to help.

Because service is what we're all about.

VISA MasterCard

800-343-5504

In Mass. Call 617-491-2700

Systems Customized / **COMPUMART** 43

65 BENT STREET, DEPT. 1306, P.O. Box 568
CAMBRIDGE, MA. 02139

TALK IS CHEAP

LET THE OTHERS TALK ABOUT PROPOSED SOLUTIONS. COMPUPRO DELIVERS RESULTS.

When you want no-excuses operation, **CompuPro** stands behind you with a family of cost-effective - and technologically innovative - products that conform fully to the IEEE 696/S-100 standards. High speed operation frees you from obsolescence as CPU clock speeds increase, while low power consumption saves energy and promotes reliability.

There's a time for toys and home entertainment computers; and there's a time for professional machines that are expandable, modular, and exceptionally reliable. When that time comes, **CompuPro** delivers the results you need.

AND NOW. . .
THE "BIG 8" SPECIAL PACKAGE!

Our **"Big 16"** package promotion went over so well that we decided to come up with something equally potent for 8 bit machines. Whether you're upgrading an existing system or assembling a brand new machine, the **"Big 8"** is ideal - just add terminal, disk drive, enclosure, and printer and you're up and running with one of the fastest, most powerful 8 bit systems around.

This easily upgradable package includes:

- **Disk 1** DMA Disk Controller
- **CPU Z** with 6 MHz CPU
- **Interfacer 1** or **2** (your choice; Interfacer 1 standard)
- **CP/M*-80** 2.2 on disk
- **32K** of fast, low power, static RAM

To sweeten the deal, we'll add another 32K of RAM if you order from us or your computer store before August 1, 1981. And if you need an enclosure, **Enclosure 2** (desktop version) is available with this package for only **$795** - giving you even more savings.

Total value of the package: $2712. . .but our special package price gives you the "Big 8" for $1995! Who says CompuPro S-100 speed and reliability can't be cost-competitive with home entertainment computers?

DISK 1: A SUPERB DISK CONTROLLER. A/T $495, CSC $595

This state of the art design uses properly implemented DMA **with arbitration,** allowing **Disk 1** to co-exist on the same bus with up to 15 other DMA devices. 24 bit DMA addressing capability allows disk access to a full 16 megabyte memory map.

Disk 1 transfers data independently of CPU speed for efficient operation with older 2 MHz CPUs as well as the new high speed 8086s; handles up to four 8" or 5.25" floppy disk drives (including 96 track high density minifloppies), single or double sided, single or double density (soft sectored); includes BIOS for CP/M-80*, as well as on-board boot for automatic startup and on-board 3 wire serial interface for system initialization; and is compatible with MP/M*, OASIS*, CP/M-80, and CP/M-86.

We weren't going to put out another me-too disk controller. . .and we didn't. Want proof? The manual is available separately for $20. The CompuPro Disk Controller is here.

COMPUTER ENCLOSURE 2
$825 desk top version, $895 rack mount version

Includes fused, constant voltage power supply (+8V at 25 Amps, +16V at 3 Amps, and -16V at 3 Amps); 20 slot shielded/active terminated motherboard; and rugged all-metal enclosure with AC outlets on rear, heavy-duty line filter, circuit breaker, low noise fan, and reset switch. Rack mount version includes slides for easy pull-out from rack frame.

Also available: COMPUTER ENCLOSURE 1. Same as above, but less power supply and motherboard. **$289** desktop, **$329** rack mount.

SYSTEM SUPPORT 1
$295 Unkit, $395 A/T, $495 CSC

Includes sockets for 4K of extended address EPROM or RAM (2716 pinout) with one battery backup RAM socket; battery operated month/day/year/time crystal clock with BCD outputs; socket for optional math processor (9511 or 9512); full RS-232 serial port; three 16 bit interval timers; two interrupt controllers; power fail indicator; and comprehensive owner's manual with numerous software examples (**manual** available separately for **$20**; add **$195** to the above prices for the optional **9512** math processor.)

*LEGAL CORNER: Z80A is a registered trademark of Zilog; PASCAL/M is a trademark of Sorcim; CP/M and MP/M are registered trademarks of Digital Research; OASIS is a trademark of Phase 1.

SOFTWARE

8088/8086 MONITOR-DEBUGGER: Supplied on single sided, single density, soft sectored 8" disk. CP/M* compatible. Great development tool; mnemonics used in debug conform as closely as possible to current CP/M* DDT mnemonics. **$35.**

PASCAL/M* FROM SORCIM: SORCIM's PASCAL/M is the best implementation we've been able to find regardless of price - a totally standard **Wirth PASCAL/M* 8" disk** and comprehensive manual. **$175** (specify Z-80* or 8080/8085 version).

S-100 MEMORIES FROM THE MEMORY LEADER

CompuPro memories feature fully static design to eliminate dynamic timing problems, flawless DMA, full conformance to all IEEE 696/S-100 specifications, high speed operation (10 MHz), low power consumption, extensive bypassing, and careful thermal design.

	Unkit	A/T	CSC
8K RAM 2A	$159	$189	$239
16K RAM 14 (extended addressing)	$279	$349	$429
16K RAM 20-16 (extended addressing and bank select)	$319	$399	$479
24K RAM 20-24 (extended addressing and bank select)	$429	$539	$629
32K RAM 20-32 (extended addressing and bank select)	$559	$699	$799
128K RAM 21	price upon request		

NEW! 64K RAM 17. Amazingly low power in a 64K **fully static** RAM board: draws less than 2.0 Watts typical, 4.0 Watts guaranteed max! It's fast, too; no wait states with 6 MHz Z-80* CPUs, or up to 10 MHz with 8086/88 family CPUs. Uses IEEE extended addressing protocol; also, user may turn off 2K windows from EOOO to FFFF in order to accommodate memory-mapped peripherals/disk controllers. (The **CompuPro** disk controller can use the full 64K since it is not memory-mapped.)**$1095** Unkit, **$1395** A/T, **$1595** CSC. 48K version also available: **$1048** A/T, **$1198.50** CSC.

HIGH SPEED S-100 CPU BOARDS

CompuPro CPU boards meet all IEEE 696/S-100 specifications (including timing). **CPU 8085/88** uses two processors, an 8085 and 8088, to provide both 16 and 8 bit capability with a standard 8 bit bus.

8 Bit **CPU Z** (with Z80A* CPU)................**$225** Unkit, **$295** A/T (both operate at 4 MHz), **$395** CSC (with 6 MHz CPU).

8 Bit **CPU 8085** (5 MHz) ....**$325** Unkit, **$325** A/T, **$425** CSC (6 MHz)

16/8 Bit **CPU 8085/88**..........................**$295** Unkit, **$425** A/T (both operate at 5 MHz); **$525** CSC (with 6 MHz 8085, 6 MHz 8088).

OTHER S-100 BUS PRODUCTS

Interfacer 1 (dual RS-232 serial ports)........**$199** Unkit, **$249** A/T, **$324** CSC
Interfacer 2 (3 parallel + 1 serial port).......**$199** Unkit, **$249** A/T, **$324** CSC
Interfacer 3 - 5 (5 serial ports)..................**$599** A/T, **$699** CSC
Interfacer 3 - 8 (8 serial ports)..................**$699** A/T, **$849** CSC
Spectrum color graphics board................**$299** Unkit, **$399** A/T, **$449** CSC
20 slot motherboard w/ edge connectors...**$174** unkit, **$214** A/T
12 slot motherboard w/ edge connectors...**$129** unkit, **$169** A/T
6 slot motherboard w/ edge connectors.....**$89** unkit, **$129** A/T
Memory Manager Board.........................**$59** unkit, **$85** A/T, **$100** CSC
Active Terminator Board........................**$34.50** Kit, **$59.50** A/T
2708 EPROM Board (2708s not included)....**$85** Unkit, **$135** A/T, **$195** CSC
Mullen Extender Board..........................**$59** Kit, **$79** A/T
Mullen Relay/Opto-Isolator Control Board.**$129** Kit, **$179** A/T

Most **CompuPro** products are available in **Unkit** form, **Assembled/Tested,** or qualified under the high-reliability **Certified System Component** (**CSC**) program (200 hour burn-in, more). Note: Unkits are not intended for novices, as de-bugging may be required due to problems such as IC infant mortality. Factory service is available for Unkits at a flat service charge.

TERMS: Prices shown do not include dealer installation and support services. Cal res add tax. Allow at least 5% shipping; excess refunded. Orders under $15 add $2 handling. VISA® and Mastercard® orders ($25 min) call our 24 hour order desk at **(415) 562-0636.** Include street address for UPS delivery. Prices are subject to change without notice.

CompuPro™ division of GODBOUT ELECTRONICS ✓42

BOX 2355, OAKLAND AIRPORT, CA 94614 (415) 562-0636

COMPUPRO PRODUCTS ARE AVAILABLE AT FINER COMPUTER STORES WORLD-WIDE. . . CALL (415) 562-0636 FOR THE STORE NEAREST YOU.

CSC Logic Probe Kit.

$17.95

Complete, easy-to-follow instructions help make this a one-night project.

BUY 5 FOR $16.15 each

$124.95 apple clock/calendar

- Seconds, minutes, hours, day-of-week, month, date, & year.
- On board batteries with one year life.
- Uses MSM5832-crystal controlled.

APPLE CARDS

SERIAL CARD	$159.00
PARALLEL CARD	$99.00
INTEGER or APPLESOFT CARD	$189.00
PASCAL SYSTEM	$425.00
PROTO BOARD	$25.00

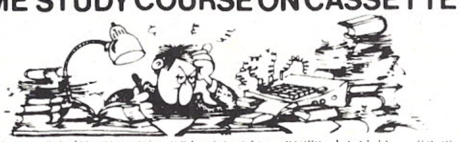
"APPLE II Plus"

48k–$1199.00
64k–$1399.00

W/PURCHASE OF APPLE RECEIVE SPECIAL PRICING:

BASE II PRINTER	$619.00
DISK II w/cont.	$595.00
DISK II w/o cont.	$495.00
SUP'R'MOD II	$23.00
MONITOR: B&W 12"–12MHz.	$125.00
INTEGER BASIC CARD	$169.00

HOME STUDY COURSE ON CASSETTE



Each course below includes a special course book plus two cassettes, total of 2½ hours. The lecture is completely coordinated to the pages of the book and can be played on any standard cassette player.

S1·INTRODUCTION TO MICROPROCESSORS

For Non-Specialists. Course contains: Definitions•Application•Evaluation Terms•System Components•2.5 hrs.

$29.95

S2·PROGRAMMING MICROPROCESSORS

For the student who has completed S-1. GOAL: To provide an over-all and practical understanding of the concepts of Micro Computer Programming. 2.5 hours.

$29.95

·NO TECHNICAL BACKGROUND ASSUMED·

78H05 $5.95 TO-3 case 5 AMPS 5 VOLTS

MSM5832 MICROPROCESSOR REAL-TIME CLOCK/CALENDAR

XTAL. $3.85

GENERAL DESCRIPTION

The MSM5832 is a monolithic metal-gate CMOS integrated circuit that functions as a real time clock calendar for use in bus oriented microprocessor applications. The on chip 32 768 Hz crystal controlled oscillator time base is counted down to provide addressable 4 bit I/O data of SECONDS, MINUTES, HOURS, DAY OF WEEK, DATE, MONTH and YEAR. Data access is controlled by 4 bit address, chip select, read, write and hold inputs. Other functions include 12H/24H format selection, leap year identification and manual ±30 second correction.

The MSM5832 normally operates from a 5 volt ±5% supply. Battery back up operation down to 2.2 volts allows continuation of time keeping when main power is off. One test input facilitates rapid testing of the time keeping operations. The MSM5832 is offered in an 18-lead dual in line plastic (RS suffix) package.

$7.45

FEATURES

- Microprocessor bus oriented
- TIME 23 59 59, MONTH 12, DATE 31, YEAR 99, DAY OF WEEK 7
- 4 BIT DATA BUS
- 4 BIT ADDRESS
- Read, Write, Hold, Chip select inputs
- Interrupt signal outputs - 1024, 1, 1/60, 1/3600 Hz
- 32 768 KHz crystal controlled operation
- Leap year register bit
- 12 or 24 hour format
- ±30 second error correction
- Single 5 volt power supply
- Back up battery operation to Vcc = 2.2 v
- Low Power Dissipation 90 μw Max. at Vcc = 3 v; 2.5 mw Max. at Vcc = 5 v
- High Density 300 mil 18 Pin Package

video 100

$129.00 ~~139.00~~ Leedex Corp 12" BLACK & WHITE LOW COST VIDEO MONITOR

COLOR

$365.00 13"

Green Phos 12 inch. $155.00

SOROC TECHNOLOGY INC.

IQ120 $689.00

No "Glitches", Surges Or Interference

The MPD 117 is the low cost solution to your power distribution problems.

$79.50

Global Specialties Logic Probes

LP1 $40

LP3 $60

2716 5V 450NS	8.50ea	8 for 7.50ea
2708 450NS	4.75ea	8 for 4.50ea
2732	25.50ea	4 for 21.50ea
4116 300NS	3.00ea	8 for 19.50
4116 200NS	3.75ea	8 for 24.50
2114L 300NS	3.65 ea	4 for 13.95
2114L 200NS	4.00ea	4 for 14.95

MODEL 800B

IMPACT PRINTER

- Dot resolution graphics in six densities
- 1920 character buffer
- Fully adjustable tractors to 9-1/2"

$649.00

base 2 inc.

UNIBOX ENCLOSURES SIX SIZES NEW

Attractive grain finish high impact ABS thermoplastic enclosures. Molded in card guides & mounting bosses provide maximum versatility. Supplied with assembly hardware.

ENCLOSURES

PART NUMBER	DIMENSIONS X		Y		Z	PRICE
110	1.25	x	2.07	x	2.77	$2.85
120	1.50	x	2.65	x	3.75	3.51
130	1.50	x	3.25	x	4.38	3.84
133	2.00	x	3.25	x	4.38	4.28
140	1.50	x	4.00	x	5.30	4.72
144	2.00	x	4.00	x	5.30	5.16

IC SOCKETS

PIN	
8	10/1.20
14	10/1.40
16	10/1.60
18	10/1.90
20	10/2.80
22	10/2.80
24	10/2.80
28	10/3.80
40	10/4.75

CPUs

Z-80	8.95
Z-80A CTC	12.50
Z-80A CPU	12.50
Z-80 002 16-64K	129.00
8085A	11.50
2901A	12.50
MC6800	9.50

New for the APPLE

MICROSOFT 16K RAM CARD $159.95

S-100 MEMORY BOARD

16K STATIC RAM

FULLY STATIC OPERATION
4K BANK ADDRESSABLE
EXTENDED MEMORY MGMT
MEETS IEEE PROPOSED S 100 SIGNAL STANDARDS
4 MHZ OPERATION

ASSEMBLED & TESTED $269.

California Computer Systems

SUPER SALE!!

Apple Expansion Kit 16K Memory Add-On Includes Instructions $25.95

TRS80 16K Add-On Instructions & Dip Switches $25.95

ATARI® 800™ COMPUTER SYSTEM

400 Computer 8K → $419.00
800 Computer 16K → $779.00

*800 COMPUTER w/48K → $989.00

*Best Buy

ATARI PERIPHERALS:

Printer ("825")	$775.00	Printer ("822")	$379.00
Recorder	$65.00	Disk Drive	$565.00
Interface (850)	$175.00	Modem	$169.00
Paddles	$17.00	Joysticks	$17.00
Star Raiders	$49.00	Assembler Editor	$49.00
Space Invaders	$17.00	Music Composer	$49.00
Chess	$32.00	Mailing List	$17.00
Kingdom	$12.00	TV Switch Box	$8.95
Hangman	$12.00	16K RAM	$155.00
Blackjack	$12.00	8K RAM	$119.00

VISICALC APPLE → 129. ATARI → 154.

"1981" I.C. Master $59.95

MB8264-20 64K RAM w/ SPEC's → $29.50

ER2051 EAROM → $4.50

P8155 → $19.50

"ALLOW 2 WKS. DEL'Y IF PERSONAL CHECK IS SENT."

concord COMPUTER PRODUCTS ✓297

1971 SO STATE COLLEGE
ANAHEIM, CALIF 92806
(714) 937-0637

VISA MasterCharge CHECK — M/O NO COD

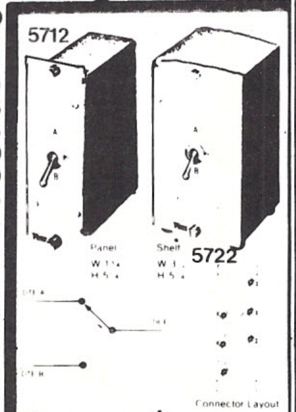
$10. MIN ORDER / CA. RES. ADD 6%

FRT.

$ 10-49	$ 2.00	$ 250-499	$ 8.00
50-99	4.00	500-999	10.00
100-24	7.00	1000-Up	Inquire

MISC.

2102 450NS	1.20
8038	2.95
NE555	.27
AY5-1013A	4.50
1488	1.25
1489	1.25
8T26	1.30
8T28	1.30
8212	2.95
8216	2.95
IS410SCR	.95
IT410TRIAC	.95
7905	.85
7908	.85
7915	.85
7918	.85
7805	.85
7806	.85
7808	.85
7812	.85
MC1330A1P	1.60
MC1350P	1.15
MC1358P	1.50
LM380	1.20
LM565N	.95
LM741	.25
MC1458P	.55
LM720	.30



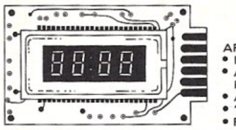
$145.00 $155.00

Description

The **RS-232C Compatible Digital Transfer Switch** is designed to switch modems between front end processors. All 24 pins of the connector are switched with Pin 1 wired to ground.

T-bar INCORPORATED

National Semiconductor Clock Modules



12VDC AUTOMOTIVE/ INSTRUMENT CLOCK

APPLICATIONS:
- In-dash autoclocks
- After-market auto/ RV clocks
- Aircraft-marine clks.
- 12VDC oper. instru.
- Portable/battery powered instrumnts.

Features: Bright 0.3" green display. Internal crystal timebase. ± 0.5 sec./day accur. Auto. display brightness control logic. Display color filterable to blue, blue-green, green & yellow. Complete—just add switches and lens.

MA1003 Module $16.95

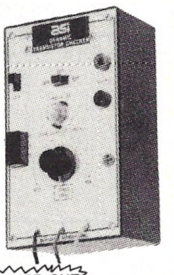
CLOCK MODULES

MA1023	.7" Low Cost Digital LED Clock Module	8.95
MA1026	.7" Dig. LED Alarm Clock/Thermometer	18.95
MA5036	.3" Low Cost Digital LED Clock/Timer	6.95
MA1002	.5" LED Display Dig. Clock & Xformer	9.95

TRANSFORMERS

XFMR1023	Xformer for MA1023 Clock Modules	3.49
XFMR1026	Xformer for MA1026 Clock Modules	3.49
XFMR5036	Xformer for MA5036 Clock Modules	3.49

Transistor Checker

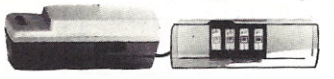


The *ASI Transistor Checker* is capable of checking a wide range of transistor types, either "in circuit" or out of circuit. It has been specially designed to take advantage of the newest concept of dynamic testing under current amplifier conditions. To operate, simply plug the transistor to be checked into the front panel socket, or connect it with the alligator clip test leads provided. No preliminary set-up is needed. The unit safely and automatically identifies low, medium and high-power PNP and NPN transistors. The ASI Transistor Checker also permits matching similar transistors in actual operating circuits, and provides a reliable GO/NO-GO test at practical collector currents (from 5mA on small signal types to 50mA and more on power types.) Match similar type transistors by observing and comparing dial readings; the higher the reading, the higher the gain. Completely assembled — battery operated. ("C" cell battery not included.)

CLOSE-OUT

Trans-Check $9.95

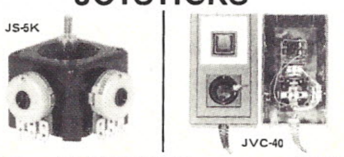
EPROM Erasing Lamp



- Erases 2708, 2716, 1702A, 5203Q, 5204Q, etc.
- Erases up to 4 chips within 20 minutes.
- Maintains constant exposure distance of one inch.
- Special conductive foam liner eliminates static build-up.
- Built-in safety lock to prevent UV exposure.
- Compact — only 7-5/8" x 2-7/8" x 2"
- Complete with holding tray for 4 chips.

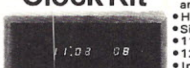
UVS-11E $79.95

JOYSTICKS



JS-5K	5K Linear Taper Pots	$5.25
JS-100K	100K Linear Taper Pots	$4.95
JVC-40	40K (2) Video Controller in case ...	$5.95

6-Digit Clock Kit



- Bright .300 ht. comm. cathode display
- Uses MM5314 clock chip
- Switches for hours, minutes and hold modes
- Hrs. easily viewable to 20 ft.
- Simulated walnut case
- 115 VAC operation
- 12 or 24 hr. operation
- Incl. all components, case & wall transformer
- Size: 6¾" x 3-1/8" x 1¾"

JE701............... $19.95

NEW! JE215 Adjustable Dual Power Supply

General Description: The JE215 is a Dual Power Supply with independent adjustable positive and negative output voltages. A separate adjustment for each of the supplies provides the user unlimited applications for IC current voltage requirements. The supply can also be used as a general all-purpose variable power supply.

FEATURES:
- Adjustable regulated power supplies, pos. and neg. 1.2VDC to 15VDC.
- Power Output (each supply): 5VDC @ 500mA, 10VDC @ 750mA, 12VDC @ 500mA, and 15VDC @ 175mA.
- Two, 3-terminal adj. IC regulators with thermal overload protection.
- Heat sink regulator cooling
- LED "on" indicator
- Printed Board Construction
- 120VAC input
- Size: 3-1/2"w x 5-1/16"L x 2"H

JE215 Adj. Dual Power Supply Kit (as shown) . . $24.95

(Picture not shown but similar in construction to above)

JE200 Reg. Power Supply Kit (5VDC, 1 amp) . .	$14.95
JE205 Adapter Brd. (to JE200) ±5, ±9 & ±12V.	$12.95
JE210 Var. Pwr. Sply. Kit, 5-15VDC, to 1.5amp.	$19.95

MICROPROCESSOR COMPONENTS

8080A/8080A SUPPORT DEVICES

INS8080A	CPU	4.95
DP8212	8-Bit Input/Output	3.25
DP8214	Priority Interrupt Control	5.95
DP8216	Bi-Directional Bus Driver	3.49
DP8224	Clock Generator/Driver	3.95
DP8226	Bus Driver	3.49
DP8228	System Controller/Bus Driver	4.95
DP8238	System Controller	5.95
INS8243	I/O Expander for 48 Series	9.95
INS8250	Asynchronous Comm. Element	16.95
DP8251	Prog. Comm. I/O (USART)	7.95
DP8253	Prog. Interval Timer	14.95
DP8255	Prog. Peripheral I/O (PPI)	9.95
DP8257	Prog. DMA Control	19.95
DP8259	Prog. Interrupt Control	14.95
DP8275	Prog. CRT Controller	49.95
DP8279	Prog. Keyboard/Display Interface	19.95
DP8300	Octal Bus Receiver	6.95
DP8303	System Timing Element	6.95
DP8304	8-Bit Bi-Directional Receiver	3.95
DP8307	8-Bit Bi-Directional Receiver	3.95
DP8308	8-Bit Bi-Directional Receiver	3.95

6800/6800 SUPPORT DEVICES

MC6800	MPU	14.95
MC6802CP	MPU with Clock and RAM	19.95
MC6810API	128x8 Static RAM	4.95
MC6821	Peripheral Inter. Adapt (MC6820)	7.49
MC6828	Priority Interrupt Controller	10.95
MC6830L8	1024x8-Bit ROM (MC68A30-8)	14.95
MC6850	Asynchronous Comm. Adapter	6.95
MC6852	Synchronous Serial Data Adapter	6.95
MC6860	0-600bps Digital MODEM	10.95
MC6862	2400bps Modulator	12.95
MC6880A	Quad 3-State Bus. Trans. (MC8T26)	2.25

MICROPROCESSOR CHIPS

Z80 (780C)	CPU (MK3880N) (2MHz)	13.95
Z80A (780-1)	CPU (MK3880N-4) (4MHz)	15.95
CDP1802	CPU	19.95
2650	MPU	16.95
IDM2901ADC	CPU—4-Bit Slice (Com. Temp. Grade)	19.95
MCS6502	MPU w/Clock (65K Bytes Memory)	11.95
INS8035N-6	MPU—8-Bit (6MHz)	16.95
INS8039N-6	CPU—Sgl. Chip 8-Bit (128 bytes RAM)	19.95
INS8040N-6	CPU (256 Bytes RAM)	24.95
INS8070N	CPU—64 Bytes RAM	24.95
INS8073N	CPU w/Basic Micro Interpreter	29.95
P8085	CPU	19.95
INS8900	CPU—16-Bit	29.95
TMS9900JL	MPU—16-Bit	49.95

SHIFT REGISTERS

MM500H	Dual 25-Bit Dynamic	.50
MM503H	Dual 50-Bit Dynamic	.50
MM506H	Dual 100-Bit Static	.50
MM510H	Dual 64-Bit Accumulator	.50
MM1402	256-Bit Dynamic	3.95
MM5013	1024-Bit Dynamic/Accumulator	1.95
MM5016H	500/512-Bit Dynamic	.89
MM5034N	Octal 80-Bit	9.95
MM5035N	Octal 80-Bit	9.95
2504V (1404A)	1024-Bit Dynamic	3.95
2518N	Hex 32-Bit Static	4.95
2522V	Dual 132-Bit Static	2.95
2524V	512-Bit Dynamic	.99
2525V	1024-Bit Dynamic	2.95
2527V	Dual 256-Bit Static	2.95
2528V	Dual 250-Bit Static	4.00
2529V	Dual 240-Bit Static	4.00
2532N	Quad 80-Bit Static	2.95
3341PC	Fifo (Dual 80)	6.95

DATA ACQUISITION

AF100-1CN	Universal Active Filter 2.5%	5.95
AF121-1CJ	Touch Tone Low Pass Filter	19.95
AF122-1CJ	Touch Tone Low Pass Filter	19.95
LM308AH	Super Gain Op Amp	1.00
LM334Z	Constant Current Source	1.30
LM335Z	Temperature Transducer	1.40
LF356N	JFET Input Op Amp	1.10
LF398N	Sample & Hold Amplifiers	3.95
LM399H	Temp. Comp. Prec. Ref. (.5ppm/C°)	4.95
ADC0804LCN	8-Bit A/D Converter (1 LSB)	4.95
DAC0806LCN	8-Bit D/A Converter (0.78% Lin.)	2.25

DATA ACQUISITION (CONTINUED)

ADC0809CCN	8-Bit A/D Converter (8-Ch. Multi.)	5.25
ADC0817CCN	8-Bit A/D Converter (16-Ch. Multi.)	10.95
DAC1000LCN	10-Bit D/A Conv. Micro. Comp. (0.05%)	13.95
DAC1008LCN	10-Bit D/A Conv. Micro. Comp. (0.20%)	8.95
DAC1020LCN	10-Bit D/A Converter (0.05% Lin.)	8.49
DAC1022LCN	10-Bit D/A Converter (0.20% Lin.)	5.95
DAC1222LCN	12-Bit D/A Converter (0.20% Lin.)	9.95
CD4051N	8-Channel Multiplexer	1.19
AY-5-1013	30K BAUD UART	5.95

RAM'S

1101	256x1 Static	1.49
1103	1024x1 Dynamic	.99
2101 (8101)	256x4 Static	3.95
2102	1024x1 Static	1.75
21L02	1024x1 Static	1.95
2111 (8111)	256x4 Static	3.95
2112	256x4 Static MOS	4.95
2114	1024x4 Static 450ns	3.95
2114L	1024x4 Static 450ns Low Power	4.95
2114-3	1024x4 Static 300ns	4.95
2114L-3	1024x4 Static 300ns Low Power	5.95
2117	16,384x1 Dynamic 550ns (house marked)	4.95
4116 (UPD416)	16K Dynamic 250ns	3.95
4164	64K Dynamic 250ns	49.95
MM2147J	4096x1 Fast 70ns	19.95
5101	256x4 Static	7.95
MM5261	1024x1 Dynamic Fully Decoded	.99
MM5262	2Kx1 Dynamic	.25
MM5280/2107	4096x1 Dynamic	4.95
MM5290J-2(4116)	16K Dynamic 150ns	5.25
MM5298J-3A	8K Dyn. 200ns (lower ½ of MM5290J)	4.95
UPD414/MK4027	4K Dynamic 16-pin	4.95
TMS4044-45NL	4K Static	14.95
TMS4045	1024x4 Static	14.95

PROMS/EPROMS

1702A	2K UV Erasable PROM	5.95
2708	8K EPROM	5.95
TMS2716	16K EPROM (-5V, +5V, +12V)	19.95
2716 Intel (2516) TI	16K EPROM (Single +5V)	10.95
2732 Intel (2532) TI	32K EPROM	19.95
2758	8K EPROM (450ns) (Single +5V)	7.95
5203	2048 PROM	14.95
82S23(74S188)	32x8 PROM (Open Collector)	4.95
82S115	4096 Bipolar PROM	19.95
82S123(74S288)	32x8 Tri-State Bipolar PROM	4.95
82S185	8K PROM	29.95

ROM'S

2513(2140)	Character Generator (Upper Case)	9.95
2513(3021)	Character Generator (Lower Case)	9.95
2516N	Character Generator	10.95
MM5230N	2048-Bit Read Only Memory	1.95

NMOS READ ONLY MEMORIES

MCM66710P	128x9x7 ASCII Shifted w/Greek	13.50
MCM66740P	128x9x7 Math Symbol & Pictures	13.50
MCM66750P	128x9x7 Alpha. Control Char. Gen.	13.50

MICROPROCESSOR MANUALS

M-Z80	User Manual	7.50
M-CDP1802	User Manual	7.50
M-2650	User Manual	5.00

SPECIAL FUNCTION

DS0025CN	Dual MOS Clock Driver (5MZ)	3.50
DS0026CN	Dual MOS Clock Driver (5MZ)	1.95
INS1771N-1	Floppy Disc Controller	24.95
INS2651N	Communication Chip	19.95
MM58167N	Microprocessor Real Time Clock	8.95
MM58174N	Microprocessor Compatible Clock	11.95
COP402N	Microcontroller with 64-Digit RAM and Direct LED Drive	6.95
COP402MN	Microcontroller with 64-Digit RAM & Direct LED Drive w/N Buss Int.	7.49
COP470N	32-Seg. VAC Fluor. Driver (20-pin pkg.)	3.25

TELEPHONE/KEYBOARD CHIPS

AY-5-9100	Push Button Telephone Dialer	14.95
AY-5-9200	Repertory Dialer	14.95
AY-5-9500	CMOS Clock Generator	4.95
AY-5-2376	Keyboard Encoder (88 keys)	11.95
HD0165-5	Keyboard Encoder (16 keys)	7.95
74C922	Keyboard Encoder (16 keys)	5.49
74C923	Keyboard Encoder (20 keys)	5.75
MM53190N	Push Button Pulse Dialer	7.95
MM57499N	96/144-Key Serial Keyboard Encoder	8.95

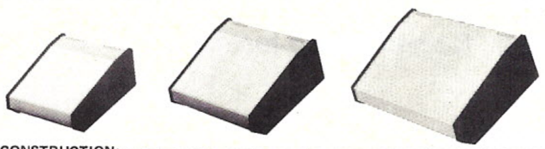
ELECTRONIC TOY MOTORS

.937 O.D. X 1.201 Length

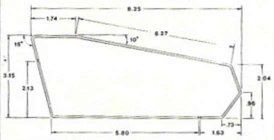
Typical Power Source	Operating Voltage Range	TYPICAL CHARACTERISTICS								SMALL TORQUE
			NO LOAD		AT MAXIMUM EFFICIENCY					
		Voltage	Speed RPM	Current AMP	Speed RPM	Current AMP	Torque OZ. IN.	Output W	Eff. %	OZ. IN.
DRY CELL	1.5-6.0	3.0	9,200	0.20	6,750	0.90	0.260	1.30	57.0	0.97

MABUCHI RE280 $ .99 each . . . 10/$7.50 . . . 100/$50.00

DESIGNERS' SERIES
Blank Desk-Top Electronic Enclosures



CONSTRUCTION:
The "DTE" Blank Desk Top Electronic Enclosures are designed to blend and complement today's modern computer equipment and can be used in both industrial and home. The end pieces are precision molded with an internal slot (all around) to accept both top and bottom panels. The panels are then fastened to ¼" thick tabs inside the end pieces to provide maximum rigidity to the enclosure. For ease of equipment servicing, the rear/bottom panel slides back on slotted tracks while the rest of the enclosure remains intact. Different panel widths may be used while maintaining a common profile outline. The molded end pieces can also be painted to match any panel color scheme.



Enclosure Model No.	Panel Width	PRICE
DTE-8	8.00"	$29.95
DTE-11	10.65"	$32.95
DTE-14	14.00"	$34.95

$10.00 Min. Order — U.S. Funds Only
Calif. Residents Add 6% Sales Tax
Postage — Add 5% plus $1 Insurance

✓41

Spec Sheets — 25¢
Send 52¢ Postage for your FREE 1981 JAMECO CATALOG

NEW 1981 CATALOG

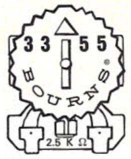
Jameco ELECTRONICS

PHONE ORDERS WELCOME (415) 592-8097

MAIL ORDER ELECTRONICS — WORLDWIDE
1355 SHOREWAY ROAD, BELMONT, CA 94002
PRICES SUBJECT TO CHANGE

6/81

Bourns Potentiometer



3/4 Watt Single Turn
(TOP ADJUSTMENT)

Values: 500Ω 1K 2.5K 5K 10K 25K 50K 100K 250K 500K 5Meg

INDIVIDUAL PRICING:

1-49	50-99	100-999	1K-up
.19	.17	.15	.12

GB174 ...... $1.95/lot
(25 pieces all 11 asst. values)

To order: Specify Bourns 3355 — (Value desired)

AC and DC Wall Transformers

Ideal for use with clocks, games, power supplies or any other type of AC or DC application.

Part No.	Input	Output	Price
AC 250	117V/60Hz	12 VAC 250mA	$3.95
AC 500	117V/60Hz	12 VAC 500mA	$4.95
AC1000	117V/60Hz	12 VAC 1 amp	$5.95
AC1700	117V/60Hz	9 VAC 1.7 amp	$6.95
DV 9200	120V/60Hz	9 VDC 200mA	$3.25
DC 900	120V/60Hz	9 VDC 500mA	$3.95

CONNECTORS

DB25P	D-Subminiature Plug	$2.95
DB25S	D-Subminiature Socket	$3.50
DB51226	Cover for DB25P/S	$1.75
22/44SE	P.C. Edge (22/44 Pin)	$2.95
UG88/U	BNC Plug	$1.79
UG89/U	BNC Jack	$3.79
UG175/U	UHF Adapter	$ .49
SO239	UHF Panel Recp.	$1.29
PL258	UHF Adapter	$1.60
PL259	UHF Plug	$1.60
UG260/U	BNC Plug	$1.79
UG1094/U	BNC Bulkhead Recp.	$1.29

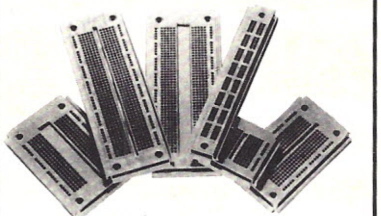
TRS-80 16K Conversion Kit

Expand your 4K TRS-80 System to 16K.
Kit comes complete with:
★ 8 ea. MM5290 (UPD416/4116) 16K Dyn. Rams (*NS)
★ Documentation for Conversion

TRS-16K2 *150NS **$39.95**
TRS-16K4 *250NS **$29.95**

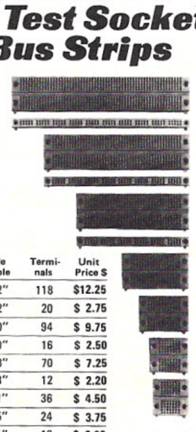
GLOBAL SPECIALTIES

EXPERIMENTOR SOCKETS



Model	Length	Width	Center Channel	5 Tie Point Terminals	Bus Strips	Price
EXP4B	6.0"	1.0"	n/a	n/a	4(160)	$ 4.75
EXP300	6.0"	2.1"	.3"	94(470)	2(80)	$12.00
EXP325	1.8"	2.1"	.3"	22(110)	2(20)	$ 3.50
EXP350	3.6"	2.1"	.3"	46(230)	2(40)	$ 6.75
EXP600	6.0"	2.4"	.6"	94(470)	2(80)	$14.75
EXP650	3.6"	2.4"	.6"	46(230)	2(40)	$ 8.75

Quick Test Sockets & Bus Strips

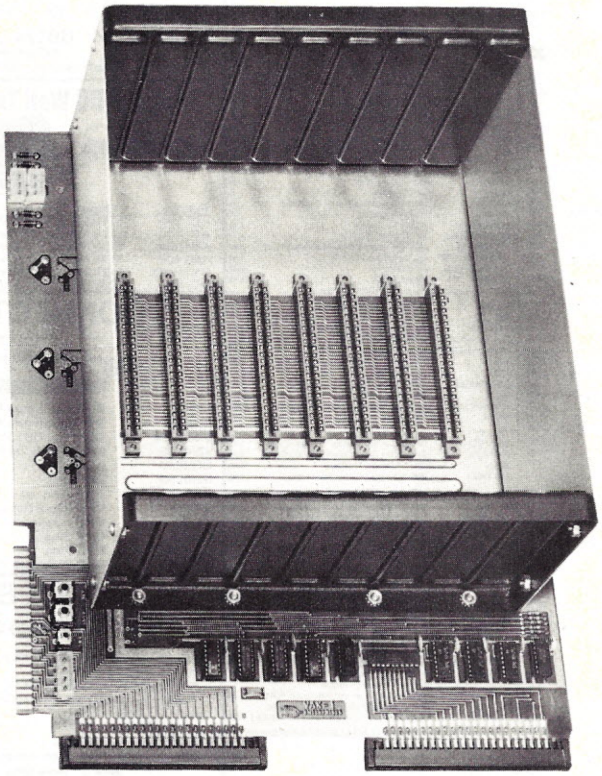


MANY OTHER GLOBAL ITEMS

	Length	Hole to-hole	Terminals	Unit Price $
QT-59S	6.5"	6.2"	118	$12.25
QT-59B	6.5"	6.2"	20	$ 2.75
QT-47S	5.3"	5.0"	94	$ 9.75
QT-47B	5.3"	5.0"	16	$ 2.50
QT-35S	4.1"	3.8"	70	$ 7.25
QT-35B	4.1"	3.8"	12	$ 2.20
QT-18S	2.4"	2.1"	36	$ 4.50
QT-12S	1.8"	1.5"	24	$ 3.75
QT-8S	1.4"	1.1"	16	$ 3.00
QT-7S	1.3"	1.0"	14	$ 2.75

✓ See List of Advertisers on page 210

VAK-1 MOTHERBOARD



We also carry:

SYM-1	**$229.00**
AIM-65 w/1K	**389.00**
AIM-65 w/4K	**439.00**

We also do custom hardware and software for the 6502 microprocessor

PRICE: **$139.00**

Call or write for shipping charges and our complete catalog.

The VAK-1 was specifically designed for use with the KIM-1, SYM-1 and the AIM 65 Microcomputer Systems. The VAK-1 uses the KIM-4* Bus Structure, because it is the only popular Multi-Sourced bus whose expansion boards were designed specifically for the 6502 Microprocessor.

SPECIFICATIONS:

- Complete with rigid CARD-CAGE
- Assembled (except for card-cage). Burned in and tested.
- All IC's are in sockets
- Fully buffered address and data bus
- Uses the KIM-4* Bus (both electrical Pin-out and card size) for expansion board slots
- Provides 8 slots for expansion boards on 1″ centers to allow for wire-wrap boards
- Designed for use with a Regulated Power Supply (such as our VAK-EPS) but has provisions for adding regulators for use with an unregulated power supply.
- Provides separate jacks for one audio-cassette, TTY and Power Supply.
- Board size: 14.5 in. Long x 11.5 in. Wide x 8 in. High
- Power requirements; 5V.DC @ 0.2 Amps.

**KIM-4 is a product of MOS Technology/C.B.M.*

RNB ENTERPRISES INCORPORATED ✓52

2951 W. Fairmount Avenue
Phoenix, AZ 85017
(602) 265-7564

master charge VISA

16K MEMORY — $24.00

FOR APPLE - TRS-80 - EXIDY - S100

4116 EQUIVALENT			2114-300ns	
NATIONAL	MM5290N-3 .... 200ns 8 for $ 24.00		NATIONAL	MM2114N-3L .... 8 for $ 26.00

FULCRUM COMPUTER PRODUCTS

IMSAI COMPATIBLE PRODUCTS

FULCRUM COMPUTER PRODUCTS

I-8080 S-100 ENCLOSURE Sheet Metal Kit — Just like THE ORIGINAL IMSAI: Mainframe with blue cover, cardguides and hardware spaced for PS-28D Power Supply, up to 22 slot motherboard.
- *Kit of all metal parts and hardware with documentation* **$115.00**
- *Thinker Toys WunderBuss 20 for above w/o conn.* .... **$ 79.00**
- *AMP S-100 Connectors—each* ... **$ 3.50**
- *8015 Blank jump-start panel w/3 switches* .... **$ 32.50**
- *8035 Jump start panel for 2 SA-400* **$ 78.50**

PS-28D Power Supply Parts Kit: — Mounts in the I-8080 enclosure, supplies +8V @ 28A, +/− 16V @ 3A, kit includes board, transformer, documentation, and all components. Improved from original.
- *Kit* .... **$ 95.50**

PIO 4-4 — 4 parallel inputs and outputs (8212) ... **$160.00**

SIO 2-2 — 2 serial I/O ports, good to 19,200 baud . **$175.00**

DIO-C/D — 2 board disk controller for 8" or 5¼" .. **$350.00**

CPM® 2.2 — For DIO including documentation **$175.00**

CPA — Improved Imsai style front panel works with Z80, etc. **$225.00**

MPU-A — 8080 processor board—requires CPA . **$100.00**

MPU-B — 8085 3MHz processor SBC w/serial plus parallel port, monitor **$250.00**

RAM III 64K MEMORY — 64K byte dynamic RAM board—Utilizes the Intel 3242 refresh controller and a single delay line for totally internal refresh. Uses time proven 4116 RAMS. Memory mapped I/O boards are allowed to coexist by the use of A16 buss pin 16.
- *Assembled & Tested* **$350.00**
- *Bare Board w/docs* .... **$ 40.00**

IKB-1 — Intelligent keyboard uses 8035 ...... **$149.50**

MDX — Dual SA400 drive enclosure .... **$ 75.00**

DE 8 — Dual 800R/801R horizontal style enclosure w/power supply and fan **$240.00**
- *Case Only* **$100.00**

COMPLETE SYSTEMS

I-8080 SYSTEM — The basic 8080 based S-100 system. Includes CPA front panel, 22 slsot motherboard (with all 22 connectors), MPU-A 8080 CPU board, PS-28D power supply (+8V @ 28A, +/−16V @ 3A), and chassis. COMPLETELY ASSEMBLED & TESTED.
- *With MPU-A* **$650.00**
- *Without MPU-A* **$600.00**
- *Thinker Toys 10MHz WunderBuss add* .... **$ 75.00**

I-8015 Complete System w/MPU-B — The complete 8085 system, includes MPU-B, RAM III, 10 slot terminated motherboard, PS-28D, and jump start front panel. A complete 64K system!
- *Assembled & Tested* **$1250.00**

I-8035 — The complete 8085 system w/2 each TANDON TM-100, DIO-D, MPU-B, RAM III, chassis, 10 slot motherboard and power supply. Includes CPM® 2.2.
- *Assembled & Tested* **$2295.00**

VDP-40 — Desk-top 8085 micro-computer system with keyboard, 9" CRT display, 10 slot S-100 board, disk controller, 64K dynamic RAM, 2 each TANDON 5¼" disk drives, 28 amp power supply.
- *Assembled & Tested* **$2895.00**

DS-8 — Dual 801R horizontal style 8" disk enclosure w/power supply, fan, and 2 Shugart 801R drives.
- *Assembled & Tested* **$1100.00**
- *Above w/DIO-C & CPM 2.2* **$1500.00**

Ask about documentation, repair service, firmware and software for your system.

component supply, inc.

1771 Junction Avenue
San Jose, California 95112
(408) 295-7171

285

TERMS: (1) PREPAID Send check or M.O. for merchandise amount only - we pay the shipping within U.S. only.
(2) UPS COD or Bankcard orders by phone or mail - shipping charges added.
California Residents add 6.5% Sales Tax.

"THE BIG BOARD"

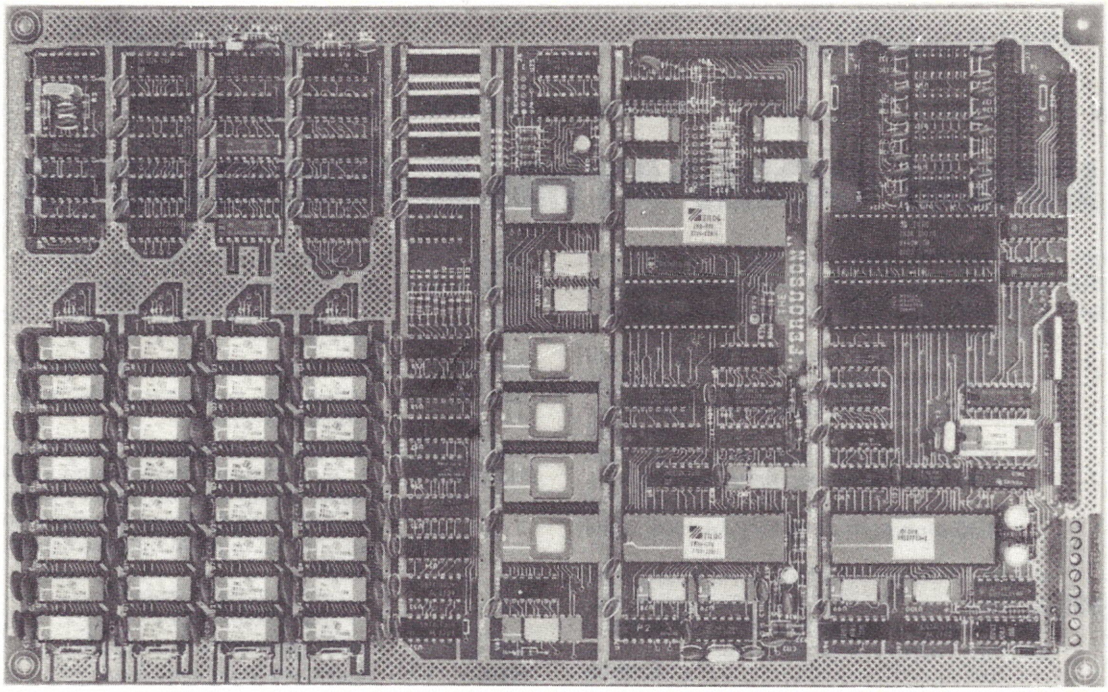
NEW! NEW!

OEM - INDUSTRIAL - BUSINESS - SCIENTIFIC

SINGLE BOARD COMPUTER KIT!

Z-80 CPU! 64K RAM!

PARTIALLY ASSEMBLED KITS
For All Sockets Installed
And Soldered Add $50.



U.K. AND EUROPE: CONTACT MACLIN-ZAND LTD. 38 MOUNT PLEASANT — LONDON WCIX OAP

THE FERGUSON PROJECT: Three years in the works, and maybe too good to be true. A tribute to hard headed, no compromise, high performance, American engineering! The Big Board gives you all the most needed computing features on one board at a very reasonable cost. The Big Board was designed from scratch to run the latest version of CP/M*. Just imagine all the off-the-shelf software that can be run on the Big Board without any modifications needed! Take a Big Board, add a couple of 8 inch disc drives, power supply, an enclosure, C.R.T., and you have a total Business System for about 1/3 the cost you might expect to pay.

$649.00 ** (64K KIT BASIC I/O)

SIZE: 8½ x 13¾ IN.
SAME AS AN 8 IN. DRIVE.
REQUIRES: +5V @ 3 AMPS
+ - 12V @ .5 AMPS.

FULLY SOCKETED!

FEATURES: (Remember, all this on one board!)

64K RAM

Uses industry standard 4116 RAM'S. All 64K is available to the user, our VIDEO and EPROM sections do not make holes in system RAM. Also, very special care was taken in the RAM array PC layout to eliminate potential noise and glitches.

Z-80 CPU

Running at 2.5 MHZ. Handles all 4116 RAM refresh and supports Mode 2 INTERUPTS. Fully buffered and runs 8080 software.

SERIAL I/O (OPTIONAL)

Full 2 channels using the Z80 SIO and the SMC 8116 Baud Rate Generator. FULL RS232. For synchronous or asynchronous communication. In synchronous mode, the clocks can be transmitted or received by a modem. Both channels can be set up for either data-communication or data-terminals. Supports mode 2 Int. Price for all parts and connectors: $85.

BASIC I/O

Consists of a separate parallel port (Z80 PIO) for use with an ASCII encoded keyboard for input. Output would be on the 80 x 24 Video Display.

24 x 80 CHARACTER VIDEO

With a crisp, flicker-free display that looks extremely sharp even on small monitors. Hardware scroll and full cursor control. Composite video or split video and sync. Character set is supplied on a 2716 style ROM, making customized fonts easy. Sync pulses can be any desired length or polarity. Video may be inverted or true. 5 x 7 Matrix - Upper & Lower Case

FLOPPY DISC CONTROLLER

Uses WD1771 controller chip with a TTL Data Separator for enhanced reliability. IBM 3740 compatible. Supports up to four 8 inch disc drives. Directly compatible with standard Shugart drives such as the SA800 or SA801. Drives can be configured for remote AC off-on. Runs CP/M* 2.2.

TWO PORT PARALLEL I/O (OPTIONAL)

Uses Z-80 PIO. Full 16 bits, fully buffered, bi-directional. User selectable hand shake polarity. Set of all parts and connectors for parallel I/O: $29.95

REAL TIME CLOCK (OPTIONAL)

Uses Z-80 CTC. Can be configured as a Counter on Real Time Clock. Set of all parts: $14.95

SYSTEM COMPARISON

64K RAM KIT	$370.00
80 x 24 Video Kit	365.00
Floppy Disk Controller Kit	235.00
Z-80 CPU Kit	185.95
SER & PAR. I/O	129.95
S-100 Mother Board	45.00
SUB TOTAL	$1330.90

Talk about bangs per buck! The prices shown for S100 kits were taken from the July 1980 BYTE. This will give some basis for comparison between the Big Board and a similar system implementation on the S100 Buss.

CP/M* 2.2 FOR BIG BOARD

The popular CP/M* D.O.S. modified by MICRONIX SYSTEMS to run on Big Board is available for $150.00.

PC BOARD

Blank PC Board with Rom Set and Full Documentation.
$199.00

PFM 3.0 2K SYSTEM MONITOR

The real power of the Big Board lies in its PFM 3.0 on board monitor. PFM commands include: Dump Memory, Boot CP/M*, Copy, Examine, Fill Memory, Test Memory, Go To, Read and Write I/O Ports, Disc Read (Drive, Track, Sector), and Search. PFM occupies one of the four 2716 EPROM locations provided.

Z-80 is a Trademark of Zilog.

Digital Research Computers
(OF TEXAS)

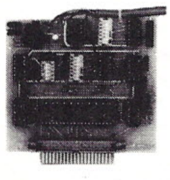
P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Shipments will be made approximately 3 to 6 weeks after we receive your order. VISA, MC, cash accepted. We will accept COD's (for the Big Board only) with a $75 deposit. Balance UPS COD. Add $3.00 shipping.

*TRADEMARK OF DIGITAL RESEARCH. NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE ORIGINATORS OF CPM SOFTWARE
**1 TO 4 PIECE DOMESTIC USA PRICE.

TRS-80 SERIAL I/O

• Can input into basic • Can use LLIST and LPRINT to output, or output continuously • RS-232 compatible • Can be used with or without the expansion bus • On board switch selectable baud rates of 110, 150, 300, 600, 1200, 2400, parity or no parity odd or even, 5 to 8 data bits, and 1 or 2 stop bits. D.T.R. line • Requires +5, -12 VDC • Board only $19.95 Part No. 8010, with parts $76.69 Part No. 8010A, assembled $98.25 Part No. 8010 C. No connectors provided, see below.



EIA/RS-232 connector Part No. DB25P $6.00, with 9, 8 conductor cable $19.65 Part No. DB25P9.

3' ribbon cable with attached connectors to fit TRS-80 and our serial board $37.10 Part No. 3CAB40.

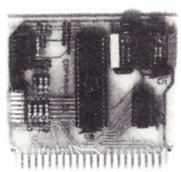
VIDEO TERMINAL

16 lines, 64 columns • Upper and lower case • 5x7 dot matrix • Serial RS-232 in and out with TTL parallel keyboard input • On board baud rate generator 75, 110, 150, 300, 600, & 1200 jumper selectable • Memory 1024 characters (7-21L02) • Video processor chip SFF96364 by Neculonic • Control characters (CR, LF, →, ←, ↑, ↓, non destructive cursor, CS, home, CL • White characters on black background or vice-versa • With the addition of a keyboard, video monitor or TV set with TV interface (part no. 107A) and power supply this is a complete stand alone terminal • also S-100 compatible • requires +16, & −16 VDC at 100mA, and 8VDC at 1A. Part No. 1000A $296.45 kit.

GAME PADDLES & SOUND FOR TRS-80

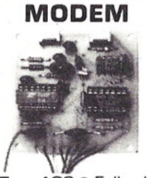
Includes: 2 game paddles, interface, software, speaker, power supply, full documentation including: schematics, theory of operation, and user guide; plus 2 games on cassette, Pong and Starship War $157.29 Complete Part No. 7922C

SERIAL/ PARALLEL INTERFACE



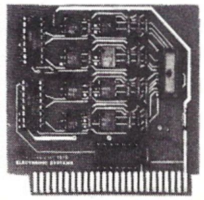
• Converts serial to parallel and parallel to serial • Low cost on board baud rate generator • 110 to 19.2K • Low power drain +5 volts and -12 volts required • TTL compatible • All characters contain a start bit, 5 to 8 data bits, 1 or 2 stop bits, and either odd or even parity. • All connections go to a 44 pin gold plated edge connector • Board only $11.95 Part No. 101, with parts $42.89 Part No. 101A, 44 pin edge connector $4.00 Part No. 44P

MODEM



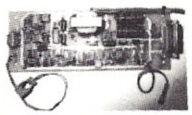
• Type 103 • Full or half duplex • Works up to 300 baud • Originate or Answer • Serial TTL input and output • connect 8 Ω speaker and crystal mic. directly to board • Requires +5 volts • Board only $7.60 Part No. 109, with parts $29.95 Part No. 109A.

OPTO-ISOLATED PARALLEL INPUT BOARD FOR APPLE II



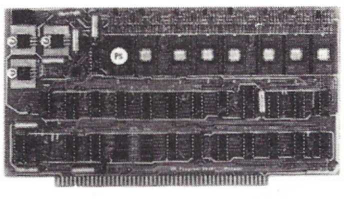
There are 8 inputs that can be driven from TTL logic or any 5 volt source. The circuit board can be plugged into any of the 8 sockets of your Apple II. It has a 16 pin socket for standard dip ribbon cable connection. Board only $15.65 Part No. 120, with parts $69.95. Part No. 120A.

SUPER MODEM



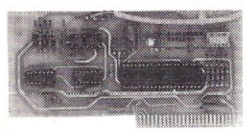
Originate, RS-232 and 20 mA compatable. Full duplex, and half duplex, direct connect or acoustic coupled, on board power supply, carrier detect light, DB25 plug, 300 BAUD, Type 103 compatable frequencies. Bare board Part No. 2000 $21.89, Kit Part No. 2000A $133.80

8K EPROM SAVER



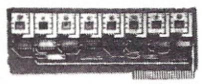
• Programs 2708's address relocation of each 4K of memory to any 4K boundary • Power on jump and reset jump option for "turnkey" systems and computers without a front panel • Program saver software in 1 2708 EPROM $25. Bare board $45.59 including custom coil, board with parts but no EPROMS $164.69.

APPLE II SERIAL I/O INTERFACE



Baud rate is continuously adjustable from 0 to 30,000 • Plugs into any peripheral connector • Low current drain. RS-232 input and output • On board switch selectable 5 to 8 data bits, 1 or 2 stop bits, and parity or no parity either odd or even • Jumper selectable address • SOFTWARE • Input and Output routine from monitor or BASIC to teletype or other serial printer • Program for using an Apple II for a video or an intelligent terminal. Also can output in correspondence code to interface with some selectrics. • Also watches DTR • Board only $14.95 Part No. 2, with parts $51.25 Part No. 2A, assembled $62.95 Part No. 2C

PARALLEL TRIAC OUTPUT BOARD FOR APPLE II



This board has 8 triacs capable of switching 110 volt 6 amp loads (660 watts per channel) or a total of 5280 watts. Board only $15.65 Part No. 210, with parts $119.95 Part No. 210A

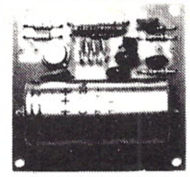
APPLE II PROTOTYPING HOBBY/CARD

Part No. 7907 $21.95

RS-232/20mA INTERFACE

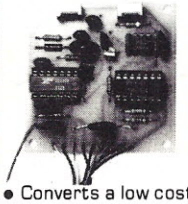
This board has two passive, opto-isolated circuits. One converts RS-232 to 20mA, the other converts 20mA to RS-232. All connections go to a 10 pin edge connector. Requires +12 and -12 volts. Board only $9.95, part no. 7901, with parts $14.95 Part No. 7901A.

T.V. INTERFACE



• Converts video to AM modulated RF, Channels 2 or 3. So powerful almost no tuning is required. On board regulated power supply makes this extremely stable. Rated very highly in Doctor Dobbs' Journal. Recommended by Apple • Power required is 12 volts AC C.T., or +5 volts DC • Board only $8.19 part No. 107, with parts $18.85 Part No. 107A

TAPE INTERFACE

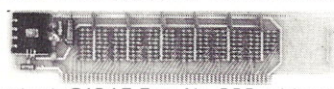


• Converts a low cost tape recorder to a digital recorder • Works up to 1200 baud • Digital in and out are TTL-serial • Output of board connects to mic. in of recorder • Earphone of recorder connects to input on board • No coils • Requires +5 volts, low power drain • Board only $7.60 Part No. 111, with parts $29.95 Part No. 111A

RS-232/TTL INTERFACE

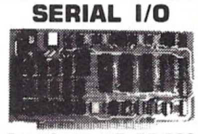
• Converts TTL to RS-232, and converts RS-232 to TTL • Two separate circuits • Requires -12 and +12 volts • All connections go to a 10 pin edge connector, kit $9.95 Part No. 232A 10 Pin edge connector $3.00 part No. 10P.

S-100 BUS ACTIVE TERMINATOR



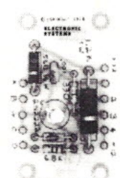
Board only $18.15 Part No. 900, with parts $29.89 Part No. 900A.

SERIAL I/O



Four Serial I/O RS-232 ports. S-100 Bus, Software or jumper selectable baud rate (110, 300, 600, 1200, 2400, 4800, 9600, 19.2K), on board Xtal baud rate generator, Addressing, switch selectable, Parity or no parity (odd or even) switch selectable, 1 or 2 stop bits, 5 to 8 bits/character. Board only $35.19 Part No. 7908. With parts (kit) $199.95, Part No. 7908A.

RS-232/TTY INTERFACE



This board has two active circuits, one converts RS-232 to 20 mA, the other converts 20 mA to RS-232. Requires +12 and -12 volts. $9.95 Part No. 600A Kit.

Send for FREE Catalog...a big self addressed envelope with 80¢ postage gets it fastest!

To Order: Mention part no., description, and price. In USA shipping paid by us for orders accompanied by check or money order. We accept C.O.D. orders (U.S. only) or a VISA or Master Charge no., expiration date, signature and phone no., shipping charges will be added. CA residents add 6.5% for tax. Outside USA add 15% for air mail postage and handling. Payment must be in U.S. dollars. Dealer inquiries invited. Prices subject to change without notice.

ORDER LINE: (408) 226-4064

ELECTRONIC SYSTEMS

Dept. KB, P.O. BOX 21638, San Jose, CA USA 95151

✓47

Start learning and computing for only $129.95 with a Netronics 8085-based computer kit. Then expand it in low-cost steps to a business/development system with 64k or more RAM, 8" floppy disk drives, hard disks and multi-terminal I/O.

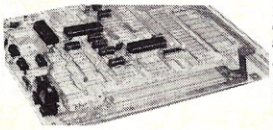
THE NEW EXPLORER/85 SYSTEM

Special! *Full 8" floppy, 64k system for less than the price of a mini! Only* **$1499.95!**

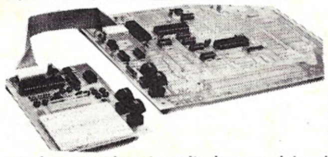
(Also available wired & tested, $1799.95)

Imagine — for only $129.95 you can own the starting level of Explorer/85, a computer that's expandable into full business/development capabilities — a computer that can be your beginner system, an OEM controller, or an IBM-formatted 8" disk small business system. From the first day you own Explorer/85, you begin computing on a significant level, and applying principles discussed in leading computer magazines. Explorer/85 features the advanced Intel 8085 cpu, which is 100% compatible with the older 8080A. It offers onboard S-100 bus expansion, Microsoft BASIC in ROM, plus instant conversion to mass storage disk memory with standard IBM-formatted 8" disks. All for only $129.95, plus the cost of power supply, keyboard/terminal and RF modulator if you don't have them (see our remarkable prices below for these and other accessories). With a Hex Keypad/display front panel, Level "A" can be programmed with no need for a terminal, ideal for a controller, OEM, or a real low-cost start.

Full 8" disk system for less than the price of a mini (shown with Netronics Explorer/85 computer and new terminal). System features floppy drive from Control Data Corp., world's largest maker of memory storage systems (not a hobby brand!)



Level "A" is a complete operating system, perfect for beginners, hobbyists, industrial controller use. $129.95



Level "A" With Hex Keypad/Display.

LEVEL "A" SPECIFICATIONS

Explorer/85's Level "A" system features the advanced Intel 8085 cpu, an 8355 ROM with 2k deluxe monitor/operating system, and an advanced 8155 RAM I/O . . . all on a single motherboard with room for RAM/ROM/PROM/EPROM and S-100 expansion, plus generous prototyping space.

PC Board: Glass epoxy, plated through holes with solder mask. • **I/O:** Provisions for 25-pin (DB25) connector for terminal serial I/O, which can also support a paper tape reader . . . cassette tape recorder input and output . . . cassette tape control output . . . LED output indicator on SOD (serial output) line . . . printer interface (less drivers) . . . total of four 8-bit plus one 6-bit I/O ports. • **Crystal Frequency:** 6.144 MHz. • **Control Switches:** Reset and user (RST 7.5) interrupt . . . additional provisions for RST 5.5, 6.5 and TRAP interrupts onboard. • **Counter/Timer:** Programmable, 14-bit binary. • **System RAM:** 256 bytes located at F800, ideal for smaller systems and for use as an isolated stack area in expanded systems . . . RAM expandable to 64K via S-100 bus or 4k on motherboard.

System Monitor (Terminal Version): 2k bytes of deluxe system monitor ROM located at FØØØ, leaving ØØØØ free for user RAM/ROM. Features include tape load with labeling . . . examine/change contents of memory . . . insert data . . . warm start . . . examine and change all registers . . . single step with register display at each break point, a debugging/training feature . . . go to execution address . . . move blocks of memory from one location to another . . . fill blocks of memory with a constant . . . display blocks of memory . . . automatic baud rate selection to 9600 baud . . . variable display line length control (1-255 characters/line) . . . channelized I/O monitor routine with 8-bit parallel output for high-speed printer . . . serial console in and console out channel so that monitor can communicate with I/O ports.

System Monitor (Hex Keypad/Display Version): Tape load with labeling . . . tape dump with labeling . . . examine/change contents of memory . . . insert data . . . warm start . . . examine and change all registers . . . single step with register display at each break point . . . go to execution address. Level "A" in this version makes a perfect controller for industrial applications, and is programmed using the Netronics Hex Keypad/Display. It is low cost, perfect for beginners.

HEX KEYPAD/DISPLAY SPECIFICATIONS

Calculator type keypad with 24 system-defined and 16 user-defined keys. Six digit calculator-type display, that displays full address plus data as well as register and status information.

LEVEL "B" SPECIFICATIONS

Level "B" provides the S-100 signals plus buffers/drivers to support up to six S-100 bus boards, and includes: address decoding for onboard 4k RAM expansion selectable in 4k blocks . . . address decoding for onboard 8k EPROM expansion selectable in 8k blocks . . . address and data bus drivers for onboard expansion . . . wait state generator (jumper selectable), to allow the use of slower memories . . . two separate 5 volt regulators.

LEVEL "C" SPECIFICATIONS

Level "C" expands Explorer/85's motherboard with a card cage, allowing you to plug up to six S-100 cards directly into the motherboard. Both cage and card are neatly contained inside Explorer's deluxe steel cabinet. Level "C" includes a sheet metal superstructure, a 5-card, gold plated S-100 extension PC board that plugs into the motherboard. Just add required number of S-100 connectors.

Explorer/85 With Level "C" Card Cage.

LEVEL "D" SPECIFICATIONS

Level "D" provides 4k of RAM, power supply regulation, filtering decoupling components and sockets to expand your Explorer/85 memory to 4k (plus the original 256 bytes located in the 8155A). The static RAM can be located anywhere from ØØØØ to EFFF in 4k blocks.

LEVEL "E" SPECIFICATIONS

Level "E" adds sockets for 8k of EPROM to use the popular Intel 2716 or the TI 2516. It includes all sockets, power supply regulator, heat sink, filtering and decoupling components. Sockets may also be used for 2k x 8 RAM IC's (allowing for up to 12k of onboard RAM).

DISK DRIVE SPECIFICATIONS

- 8" CONTROL DATA CORP. professional drive.
- LSI controller.
- Write protect.
- Single or double density.
- Data capacity: 401,016 bytes (SD), 802,032 bytes (DD), unformatted.
- Access time: 25ms (one track).

DISK CONTROLLER/ I/O BOARD SPECIFICATIONS

- Controls up to four 8" drives.
- 1771A LSI (SD) floppy disk controller.
- Onboard data separator (IBM compatible).
- 2 Serial I/O ports
- Autoboot to disk system when system reset.
- 2716 PROM socket included for use in custom applications.
- Onboard crystal controlled.
- Onboard I/O baud rate generators to 9600 baud.
- Double-sided PC board (glass epoxy.)

DISK DRIVE CABINET/POWER SUPPLY

- Deluxe steel cabinet with individual power supply for maximum reliability and stability.

ORDER A COORDINATED EXPLORER/85 APPLICATIONS PAK!

Beginner's Pak (Save $26.00!) — Buy Level "A" (Terminal Version) with Monitor Source Listing and AP-1 5-amp Power Supply: (regular price $199.95), now at SPECIAL PRICE: **$169.95** plus post. & insur.

Experimenter's Pak II (Save $53.40!) — Buy Level "A" (Hex Keypad/Display Version) with Hex Keypad/Display, Intel 8085 User Manual, Level "A" Hex Monitor Source Listing, and AP-1 5-amp Power Supply: (regular price $279.35), all at SPECIAL PRICE: **$219.95** plus post. & insur.

Special Microsoft BASIC Pak (Save $103.00!) — Includes Level "A" (Terminal Version), Level "B", Level "D" (4k RAM), Level "E", 8k Microsoft in ROM, Intel 8085 User Manual, Level "A" Monitor Source Listing, and AP-1 5-amp Power Supply: (regular price $439.70), now yours at SPECIAL PRICE: **$329.95** plus post. & insur.

ADD A TERMINAL WITH CABINET, GET A FREE RF MODULATOR: Save over $114 at this SPECIAL PRICE: **$499.95** plus post. & insur.

Special 8" Disk Edition Explorer/85 (Save over $104!) — Includes disk-version Level "A", Level "B", two S-100 connectors and brackets, disk controller, 64k RAM, AP-1 5-amp power supply, Explorer/85 deluxe steel cabinet, cabinet fan, 8" SD/DD disk drive from famous CONTROL DATA CORP. (not a hobby brand!), drive cabinet with power supply, and drive cable set-up for two drives. This package includes everything but terminal and printers (see coupon for them). Regular price $1630.30, all yours in kit at SPECIAL PRICE: **$1499.95** plus post. & insur. Wired and tested, only **$1799.95.**

Special! Complete Business Software Pak (Save $625.00!) — Includes CP/M 2.0, Microsoft BASIC, General Ledger, Accounts Receivable, Accounts Payable, Payroll Package: (regular price $1325), yours now at SPECIAL PRICE: **$699.95.**

Please send the items checked below:

- ☐ **Explorer/85 Level "A" kit** (Terminal Version) . . . **$129.95** plus $3 post. & insur.
- ☐ **Explorer/85 Level "A" kit** (Hex Keypad/Display Version) . . . **$129.95** plus $3 post. & insur.
- ☐ **8k Microsoft BASIC** on cassette tape, **$64.95** postpaid.
- ☐ **8k Microsoft BASIC in ROM kit** (requires Levels "B", "D" and "E") . . . **$99.95** plus $2 post. & insur.
- ☐ **Level "B" (S-100) kit** . . . **$49.95** plus $2 post. & insur.
- ☐ **Level "C" (S-100 6-card expander) kit** . . . **$39.95** plus $2 post. & insur.
- ☐ **Level "D" (4k RAM) kit** . . . **$69.95** plus $2 post. & insur.
- ☐ **Level "E" (EPROM/ROM) kit** . . . **$5.95** plus 50¢ p&h.
- ☐ **Deluxe Steel Cabinet** for Explorer/85 . . . **$49.95** plus $3 post. & insur.
  - ☐ Fan For Cabinet . . . **$15.00** plus $1.50 post. & insur.
- ☐ **ASCII Keyboard/Computer Terminal kit:** features a full 128 character set, u&l case; full cursor control; 75 ohm video output; convertible to baudot output; selectable baud rate, RS232-C or 20 ma. I/O, 32 or 64 character by 16 line formats, and can be used with either a CRT monitor or a TV set (if you have an RF modulator) . . . **$149.95** plus $3.00 post. & insur.
  - ☐ **DeLuxe Steel Cabinet** for ASCII keyboard/terminal . . . **$19.95** plus $2.50 post. & insur.
- ☐ **New! Terminal/Monitor:** (See photo) Same features as above, except 12" monitor with keyboard and terminal is in deluxe single cabinet; kit . . . **$399.95** plus $7 post. & insur.
- ☐ **Hazeltine terminals:** Our prices too low to quote — **CALL US**
- ☐ **Lear-Sigler terminals/printers:** Our prices too low to quote: **CALL US**
- ☐ **Hex Keypad/Display kit** . . . **$69.95** plus $2 post. & insur.
- ☐ **AP-1 Power Supply Kit** ±8V @ 5 amps) in deluxe steel cabinet . . . **$39.95** plus $2 post. & insur.
- ☐ **Gold Plated S-100 Bus Connectors** . . . **$4.85** each, postpaid.
- ☐ **RF Modulator kit** (allows you to use your TV set as a monitor) . . . $8.95 **postpaid.**
- ☐ **16k RAM kit** (S-100 board expands to 64k) . . . **$199.95** plus $2 post. & insur.
- ☐ **32k RAM kit** . . . **$299.95** plus $2 post. & insur.
- ☐ **48k RAM kit** . . . **$399.95** plus $2 post. & insur.
- ☐ **64k RAM kit** . . . **$499.95** plus $2 post. & insur.
- ☐ **16k RAM Expansion kit** (to expand any of the above in 16k blocks up to 64k) . . . **$99.95** plus $2 post. & insur. each.
- ☐ **Intel 8085 cpu Users' Manual** . . . **$7.50** postpaid.
- ☐ **12" Video Monitor** (10MHz bandwidth) . . . **$139.95** plus $5 post. & insur.
- ☐ **Beginner's Pak** (see above) **$169.95** plus $4 post. & insur.
- ☐ **Experimenter's Pak** (see above) . . . **$219.95** plus $6 post. & insur.
- ☐ **Special Microsoft BASIC Pak Without Terminal** (see above) . . . **$329.95** plus $7 post. & insur.
- ☐ Same as above, plus **ASCII Keyboard Terminal With Cabinet, Get Free RF Modulator** (see above) . . . **$499.95** plus $10 post. & insur.
- ☐ **Special 8" Disk Edition Explorer/85** (see above) . . . **$1499.95** plus $26 post. & insur.
  - ☐ **Wired & Tested** . . . **$1799.95** plus $26 post. & insur.
- ☐ **Extra 8" CDC Floppy Drives** . . . **$499.95** plus $12 post. & insur.
  - ☐ **Cabinet & Power Supply For Drive** . . . **$69.95** plus $3 post. & insur.
- ☐ **Drive Cable Set-up For Two Drives** . . . **$25** plus $1.50 post. & insur.
- ☐ **Disk Controller Board With I/O Ports** . . . **$199.95** plus $2 post. & insur.
- ☐ **Special: Complete Business Software Pak** (see above) . . . **$699.96** postpaid.

SOLD SEPARATELY:

- ☐ **CP/M 1.4** . . . **$100** postpaid.
- ☐ **CP/M 2.0** . . . **$150** postpaid.
- ☐ **Microsoft BASIC** . . . **$325** postpaid.
- ☐ **Intel 8085 cpu User Manual** . . . $7.50 **postpaid.**
- ☐ **Level "A" Monitor Source Listing** . . . **$25** postpaid.

Dept. KB6

Continental U.S.A. Credit Card Buyers Outside Connecticut

CALL TOLL FREE: 800-243-7428

To Order From Connecticut Or For Technical Assistance, call (203) 354-9375

Total Enclosed (Conn res. add sales tax) $ ______

Paid By:

☐ Personal Check ☐ Cashier's Check/Money Order

☐ VISA ☐ **Master Charge** (Bank No. ____)

Acct. No. ______ Exp. Date ______

Signature ______

Print Name ______

Address ______

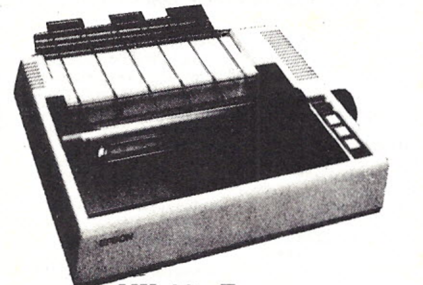
City ______

State ______ Zip ______

NETRONICS Research & Development Ltd.
333 Litchfield Road, New Milford, CT 06776

Low Prices,

Printers

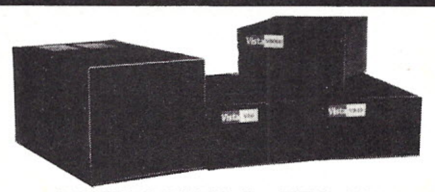


MX-80 - Epson
132 column, 9 x 9 dot matrix, multiple fonts
PRM-27080 *Save $100.00* .... **Call**

MX-70
PRM-27070 *With Graftrax II* .... **Call**
Interface & Cable for Apple **$110.00**

SPINWRITER - NEC
65 cps, bi-directional, letter quality printer with deluxe tractor mechanism, both parallel and serial interfaces on-board, 16K buffer, ribbon, print thimble, graphics, micro-space justification, data cable, and self test/diagnostic ROM.
PRD-55511 *without 16K buffer* ... **$2795.00**
PRD-55512 *with 16K buffer* **$2895.00**

Accessories for TRS-80



DISK DRIVES for TRS-80
23% more storage, 8 times faster, 40 track with free patch, 120 day warranty.
MSM-12410C *Save $125.00 !!!* **$325.00**

8″ DISK DRIVES for MODEL II
2 double density drives with cabinet, power supply, & cables
END-000433 *Kit* **$1050.00**
END-000434 *Assembled* **$1250.00**
WCA-5036A *Cable (required)* **$29.95**

Special Purchase - Save $50.00
Novation Cat Modem
300 baud, answer and originate

(We have only ~~200~~ 265 availible at this special price)

IOM-5200A *List price $189.95* **$139.95**
D-CAT *300 baud, direct connect modem*
IOM-5201A *Special sale price* **$189.00**
AUTO-CAT *Auto answer/origiate, direct connect*
IOM-5230A *Special sale price* **$239.95**

Accessories for Apple

16K MEMORY UPGRADE
Add 16K of RAM to your TRS-80, Apple, or Exidy in just minutes. We've sold thousands of these 16K RAM upgrades which include the appropriate memory chips (as specified by the manufacturer), all necessary jumper blocks, fool-proof instructions, and our 1 year guarantee.
MEX-16100K *TRS-80 kit* **$29.00**
MEX-16101K *Apple kit* **$29.00**
MEX-16102K *Exidy kit* **$29.00**

16K RAM Card - Microsoft
(There is life after 48K)
MEX-16300A *A & T* **$174.95**

Z-80* CARD for APPLE
Two computers in one, Z-80 & 6502, more than doubles the power & potential of your Apple, includes Z-80 CPU card, CP/M 2.2, & BASIC-80*
CPX-30800A *A & T* **$279.95**

Atari 800 $799.95

APPLE CLOCK - Cal Comp Sys
Real time clock w/battery back-up
IOK-2030A *A & T* **$109.95**

APPLE STICK - Micromate
Joy stick with pots for Apple II
SYA-1510A *A & T* **$49.95**

VISICALC - Personal Sftwr
The ultimate program for your Apple II
SFA-24101005M *Complete package* **$139.95**

DOS 3.3 UPGRADE - Apple
Upgrade your old DOS to the improved 3.3
IOD-2233A *Complete kit* **$64.95**

DISK DRIVE for APPLE
5¼″ disk drive with controller for your Apple
MSM-12310C *with controller* **$475.00**
MSM-123101 *w/out controller* **$375.00**

8″ DRIVES for APPLE
Controller, DOS, two 8″ double densisty drives, cabinet, power supply, & cables
Special Package Price *Kit* **$1399.95**

PRINTER INTERFACE - C.C.S.
Centronics type I/O card w/ firmware
IOI-2041A *A & T* **$99.95**

AIO, ASIO, APIO - S.S.M.
Parallel & serial interface for your Apple (see Byte pg 11)
IOI-2050K *Par & Ser kit* **$129.95**
IOI-2050A *Par & Ser A & T* **$159.95**
IOI-2052K *Serial kit* **$89.95**
IOI-2052A *Serial A & T* **$99.95**
IOI-2054K *Parallel kit* **$69.95**
IOI-2054A *Parallel A & T* **$89.95**

A488 - S.S.M.
IEEE 488 controller, uses simple basic commands, includes firmware and cable, 1 year guarantee, (see April Byte pg 11)
IOX-7488A *A & T* **$399.95**

Disk Drives

JADE's new dual disk sub-assemblies include: ***Handsome*** *metal cabinet with* ***proportionally balanced*** *air flow system,* ***assembled & tested*** *dual drive power supply,* ***quiet whisper type*** *cooling fan, power-cable kit, lighted power switch, approved fuse assembly, line cord,* ***Never-Mar*** *rubber feet, and all necessary hardware to mount 2-8″ disk drives - it's* ***all American made, guaranteed for six months****, and it's in stock!*

Dual 8″ Sub-Assembly Cabinet
END-000421 *Cabinet kit* **$225.00**
END-000420 *Bare cabinet* **$59.95**

Single sided, double density disk drive sub-system
END-000423 *Kit w/2 8″ drives* **$975.00**
END-000424 *A & T w/2 8″ drives* **$1195.00**

Double sided, double density disk drive sub-system
END-000426 *kit w/2 8″ drives* **$1495.00**
END-000427 *A & T w/2 8″ drives* **$1695.00**

8″ DISK DRIVES
Highly reliable double density floppy disk drives
Shugart 801R single sided, double density
MSF-10801R *SA-801R* **$425.00**
Special Sale Price .... ***2 for $790.00***

Shugart 851R double sided, double density
MSF-10851R *SA-851R* **$595.00**
Special Sale Price .... ***2 for $1150.00***

Siemens FDD100-8D2 single sided, double density
MSF-201120 *6 mo warranty* **$385.00**
Special sale price .... ***2 for $750.00***

Qume Datatrak 8 double sided, double density
MSF-750080 *SA-851R compatible* .. **$599.95**
Special sale price .... ***2 for $1160.00***

JADE DISK PACKAGE
Double density controller, two 8″ double density floppy disk drives, CP/M 2.2 (configured for controller), hardware and software manuals, boot PROM, cabinet, power supply, fan, & cables
Special Package Price *Kit* **$1395.00**

Diskettes

DISKETTES - Jade
Bargain prices on magnificent magnetic media
5¼″ single sided, single density, box of 10
MMD-5110103 *Soft sector* **$27.95**
MMD-5111003 *10 sector* **$27.95**
MMD-5111603 *16 sector* **$27.95**
5¼″ double sided, double density, box of 10
MMD-5220103 *Soft sector* **$39.95**
8″ single sided, single density, box of 10
MMD-8110103 *Soft sector* **$33.95**
8″ single sided, double density, box of 10
MMD-8120103 *Soft sector* **$39.95**
8″ double sided, double density, box of 10
MMD-8220103 *Soft sector* **$49.95**

Video Monitors

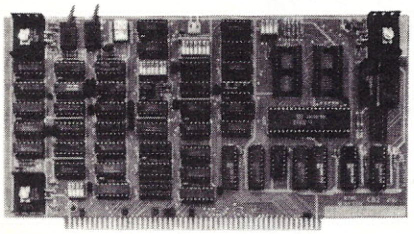
13″ COLOR MONITOR - Zenith
The hi res color you've been promising yourself
VDC-201301 .... **$449.00**

12″ GREEN SCREEN - NEC
20 MHz, P31 phosphor video monitor with audio, exceptionally high resolution - A fantastic monitor at a very reasonable price
VDM-651200 *12″ monitor* **$259.95**

Leedex / Amdek
Reasonably priced video monitors
VDM-801210 *Video 100 12″ B&W* .. **$149.95**
VDM-801230 *Video 100-80 12″ B&W* **$189.95**
VDM-801250 *12″ Green Phospor* **$189.95**
VDC-801310 *13″ Color I* **$399.95**

Fast Service.

S-100 CPU



CB-2 Z-80 CPU - S.S.M.
2 or 4 MHz Z-80 CPU board with provision for up to 8K of ROM or 4K of RAM on board, extended addressing, IEEE S-100, front panel compatible.

CPU-30300K *Kit* **$239.95**
CPU-30300A *A & T* **$299.95**

THE BIG Z* - Jade
2 or 4 MHz switchable Z-80 CPU with serial I/O, accomodates 2708, 2716, or 2732 EPROM, baud rates from 75 to 9600*

CPU-30201K *Kit* **$145.00**
CPU-30201A *A & T* **$199.00**
CPU-30200B *Bare board* **$35.00**

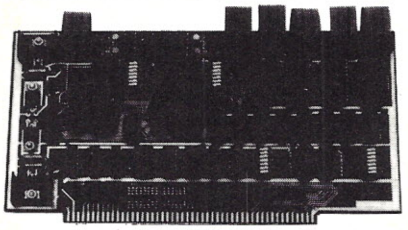
2810 Z-80* CPU - Cal Comp Sys
2/4 MHz Z-80A CPU with RS-232C serial I/O port and on-board MOSS 2.2 monitor PROM, front panel compatible.*

CPU-30400A *A & T* **$269.95**

SBC-200 - SD Systems
4 MHz Z-80 CPU with serial & parallel I/O ports, up to 8K of on-board PROM, software programmable baud rate generator, 1K of on-board RAM, Z-80 CTC.*

CPC-30200K *Kit* **$339.95**
CPC-30200A *Jade A & T* **$399.95**

S-100 I/O



I/O-4 - S.S.M.
2 serial I/O ports plus 2 parallel I/O ports

IOI-1010K *Kit* **$159.95**
IOI-1010A *A & T* **$219.95**
IOI-1010B *Bare board* **$35.00**

S.P.I.C. - Jade
Our new I/O card with 2 SIO's, 4 CTC's, and 1 PIO

IOI-1045K *2 CTC's, 1 SIO, 1 PIO* .. **$199.00**
IOI-1045A *A & T* **$259.00**
IOI-1046K *4 CTC's, 2 SIO's, 1 PIO* **$259.00**
IOI-1046A *A & T* **$319.00**
IOI-1045B *Bare board w/ manual* ... **$59.95**
IOI-1045D *Manual only* **$20.00**

Motherboards

ISO-BUS - Jade
Silent, simple, and on sale - a better motherboard

6 Slot (5¼" x 8⅝")

MBS-061B *Bare board* **$19.95**
MBS-061K *Kit* **$39.95**
MBS-061A *A & T* **$49.95**

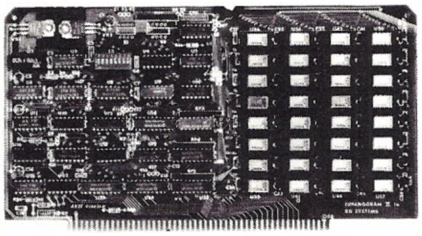
12 Slot (9¾" x 8⅝")

MBS-121B *Bare board* **$29.95**
MBS-121K *Kit* **$69.95**
MBS-121A *A & T* **$89.95**

18 Slot (14½" x 8⅝")

MBS-181B *Bare board* **$49.95**
MBS-181K *Kit* **$99.95**
MBS-181A *A & T* **$139.95**

S-100 Memory



EXPANDORAM II - S D Systems
4 MHz RAM board expandable from 16K to 64K

MEM-16630K *16K kit* **$275.95**
MEM-32631K *32K kit* **$295.95**
MEM-48632K *48K kit* **$315.95**
MEM-64633K *64K kit* **$335.95**
Assembled & tested .... add **$50.00**

64K RAM - Calif Computer Sys
4 MHz bank port / bank byte selectable, extended addressing, 16K bank selectable, PHANTOM line allows memory overlay, 8080 / Z-80 / front panel compatible.

MEM-64565A *A & T* **$575.00**

MEMORY BANK - Jade
4 MHz, IEEE S-100, bank selectable, 8 or 16 bit, expandable from 16K to 256K

MEM-99730B *Bare board* **$55.00**
MEM-99730K *Kit, no RAM* **$219.95**
MEM-16730K *16K kit* **$249.95**
MEM-32731K *32K kit* **$289.95**
MEM-48732K *48K kit* **$324.95**
MEM-64733K *64K kit* **$359.95**
Assembled & tested .... add **$50.00**

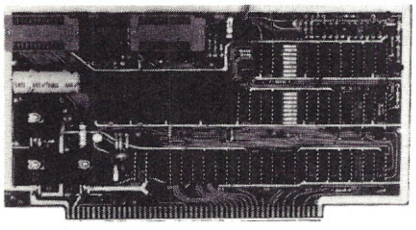
32K STATIC RAM - Jade
2 or 4 MHz expandable static RAM board uses 2114L's

MEM-16151K *16K 4 MHz kit* **$169.95**
MEM-32151K *32K 4 MHz kit* **$299.95**
Assembled & tested .... add **$50.00**

16K STATIC RAM - Cal Comp Sys
2 or 4 MHz 16K static RAM board, IEEE S-100, bank selectable, Phantom capability, addressable in 4K blocks

MEM-16160A *16K 2 MHz A & T* ... **$286.95**
MEM-16162A *16K 4 MHz A & T* ... **$289.95**
MEM-16160B *Bare board* **$50.00**

S-100 PROM Boards



PB-1 - S.S.M.
2708, 2716 EPROM board with built-in programmer

MEM-99510K *Kit* **$139.95**
MEM-99510A *A & T* **$199.95**

PROM-100 - SD Systems
2708, 2716, 2732, 2758, & 2516 EPROM programmer

MEM-99520K *Kit* **$219.95**
MEM-99520A *Jade A & T* **$269.95**

EPROM BOARD - Jade
16K or 32K uses 2708's or 2716's, 1K boundary

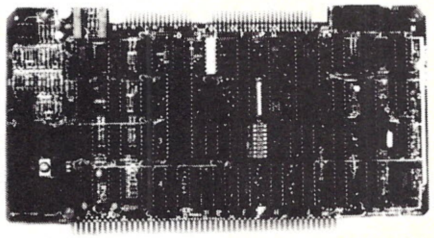
MEM-16230K *Kit* **$79.95**
MEM-16230A *A & T* **$119.95**

Mainframes

MAINFRAME - Cal Comp Sys
12 slot S-100 mainframe with 20 amp power supply

ENC-112106 A & T **$429.95**

S-100 Disk Controller



DOUBLE-D - Jade
Double density controller with the inside track, on-board Z-80A, printer port, IEEE S-100, can function on an interrupt driven buss*

IOD-1200K *Kit* **$299.95**
IOD-1200A *8" A & T* **$389.95**
IOD-1205A *5¼" A & T* **$389.95**
IOD-1200B *Bare board* **$65.00**

DOUBLE DENSITY - Cal Comp Sys
5¼" and 8" disk controller, single or double density, with on-board boot loader ROM, and free CP/M 2.2 and manual set.*

IOD-1300A *A & T* **$369.95**

VERSAFLOPPY II - SD Systems
New double density controller for both 8" & 5¼"

IOD-1160K *Kit* **$339.95**
IOD-1160A *A & T* **$379.95**

S-100 Video

VB-3 - S.S.M.
80 characters x 24 lines expandable to 80 x 48 for a full page of text, upper & lower case, 256 user defined symbols, 160 x 192 graphics matrix, memory mapped, has key board input.

IOV-1095K *4 MHz kit* **$345.00**
IOV-1095A *4 MHz A & T* **$395.95**
IOV-1096K *80 x 48 upgrade* **$39.95**

VDB-8024 - SD Systems
80 x 24 I/O mapped video board with keyboard I/O, and on-board Z-80A.*

IOV-1020K *Kit* **$399.95**
IOV-1020A *Jade A & T* **$459.95**

VIDEO BOARD - Jade
64 characters x 16 lines, 7 x 9 dot matrix, full upper/lower case ASCII character set, numbers, symbols, and greek letters, normal/reverse/blinking video, S-100.

IOV-1050K *Kit* **$99.95**
IOV-1050A *A & T* **$125.00**
IOV-1050B *Bare board* **$29.95**

2114L Low Power 4MHz

1 - 19	20 - 99	100 or more
$3.35	$2.99	$2.50

Place Orders Toll Free

Continental U.S.	*Inside California*
800-421-5500	**800-626-1710**

For Technical Inquires or Customer Service call:
213-973-7707

JADE
Computer Products ✓48

4901 W. Rosecrans, Hawthorne, Ca 90250

TERMS of SALE: Cash, checks, credit cards, or Purchase Orders from qualified firms and institutions. **Minimum Order $15.00.** California residents add 6% tax. Minimum shipping & handling charge $3.00. **Pricing & availibility subject to change**

Why use their flexible discs:

Ampex, Athana, BASF, Caelus, Control Data, Dysan, IBM, Inmac, K-Line, Maxell, Nashua, Scotch, Shugart, Syncom, 3M, Verbatim or Wabash

when you could be using

MEMOREX

for as low as $1.99 each?

Find the flexible disc you're now using on our cross reference list... then write down the equivalent Memorex part number you should be ordering.

Product Family	Product Description	Memorex Part Number (3201-)	CE quant. 100 price per disc ($)	Ampex	Athana	BASF	Caelus	Dysan	IBM	K-Line	Maxell	Nashua	Scotch 3M	Shugart	Syncom	Verbatim	Wabash	Inmac	Control Data
	IBM Compatible (128 B/S, 26 sectors)	3060	2.19	5FD-111110	473071	53428	CM-F11	800506	2305830	40013	FD1-128	FD-1	740-0	S/A 100	15002	FD34-1000	F111111X	7870-K	421602
Flexible Disc 1s	IBM Compatible (128 B/S, 26 sectors) w/ W.P.N.	3062	2.24	–	–	–	–	–	–	–	–	–	740-0	–	–	–	–	–	–
	IBM Compatible (128 B/S, 26 sectors) w/ W.P.N. & Hub ring	3064	2.55	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Single-Headed Drives	IBM Compatible (128 B/S, 26 sectors) REVERSIBLE	1729	3.35	5FD-113110	473072	54431	–	–	–	40015	–	FD-2	740/2-0	–	15150	FF34-2000	F171111X	7860-K	–
Single-Density Media	IBM System 6 Compatible	3066	2.19	–	473077	54561	–	800509	1669959	40014	–	–	740-0 0S6	–	15003	FD60-1000	F116111X	–	–
	IBM Compatible (256 B/S, 15 sectors)	3109	2.19	5FD-111210	473073	–	–	800584	2305845	40040	–	–	740-3600	–	15005	FD36-1000	F112111X	7861-K	–
	IBM Compatible (512 B/S, 8 sectors)	3110	2.19	–	473074	–	–	800585	1669954	40044	–	–	–	–	15004	FD60-1000	F113111X	7889-K	–
	Shugart Compatible, 32 hard sector	3015	2.19	5FD-211010	470901	53802	CM-F21	101/1	–	40016	FH1-32	FD-132	740-32	S/A-101	15025	FD32-1000	–	7890-K	421322
	Shugart Compatible, 32 hard sector REVERSIBLE	3025	3.35	5FD-213010	–	–	–	–	–	40017	–	–	740/2-32	–	15151	FF32-2000	–	7880-K	–
	Wang Compatible, 32 hard sector w/Hub ring	3087	2.50	–	–	54491	–	–	–	–	–	–	740-32RM	–	–	–	F37A411X	–	–
	CPT 8000 Compatible	3045	2.79	–	–	–	–	–	–	–	–	–	–	–	15226	–	–	–	–
Flexible Disc 1d	IBM Compatible (128 B/S, 26 sectors)	3090	2.95	5FD-121010	474071	54568	–	3740/1D	–	40047	FD1-128/M2100	FD-1D	741-0	–	–	FD34-8000	F131111X	7857-K	423002
	Soft Sector (128 B/S, 26 sectors) REVERSIBLE	3093	3.99	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Single-Headed Drives	Shugart Compatible, 32 hard sector	3091	2.95	5FD-221010	470801	54596	–	101/1D	–	40024	FH1-32D	–	741-32	S/A-103	15075	FD32-8000	F33A410X	7887-K	423322
Double-Density Media	Shugart Compatible, 32 hard sector REVERSIBLE	3094	3.99	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Wang Compatible, 32 hard sector w/Hub ring	3088	3.20	–	–	–	–	–	–	–	–	–	–	–	–	–	F22A411X	–	–
Flexible Disc 2s	Soft Sector (Unformatted)	3101	3.84	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Soft Sector (128 B/S, 26 sectors)	3113	3.84	–	–	54428	–	800814	1766870	–	–	–	–	S/A-150	15153	FD10-4026	F121111X	–	–
Double-Headed Drives	Soft Sector (256 B/S, 15 sectors)	3106	3.84	–	473477	54226	–	800815	2736700	40043	FD2-256D	–	742-0	–	15154	FD10-4015	F122111X	7856-K	424612
Single-Density Media	32 Hard Sector	3108	3.84	–	–	–	–	–	–	–	FH2-32	–	–	–	–	–	–	–	–
Flexible Disc 2d	Soft Sector (Unformatted)	3102	3.49	–	473485	–	–	DY150	–	40028	FD2-XDM	FD-2D	743-0	–	15103	DD34-4001	–	–	425002
	Soft Sector (128 B/S, 26 sectors)	3115	3.49	–	–	–	–	–	–	–	–	–	–	S/A-150	–	–	–	–	–
Double-Headed Drives	Soft Sector (256 B/S, 26 sectors)	3103	3.49	–	473471	54325	–	800817	1766872	40019	FD2-2560	–	743-0/256	–	15101	DD34-4026	F144111X	7858-K	425602
Double-Density Media	Soft Sector (512 B/S, 15 sectors)	3114	3.49	–	473472	54479	–	800818	1669044	40039	–	–	743-0/512	–	15100	DD34-4015	F145111X	–	425612
	Soft Sector (1024 B/S, 8 sectors)	3104	3.49	–	473473	54485	–	800819	1669045	40020	–	–	743-0/1024	–	15102	DD34-4008	F147111X	7859-K	425622
	32 Hard Sector	3105	3.49	5FD-321010	470851	–	–	101/2D	–	40021	FH2-32D	–	743-32	S/A-151	15125	DD32-4000	F34A411X	7881-K	425322
	Burroughs B-80 Compatible 32 Hard Sector	3092	3.49	–	–	–	–	–	–	–	–	–	–	–	–	–	F34A611X	–	–
	Soft Sector (1024 B/S, 8 sectors) w/ Hub Ring	3116	3.75	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Flexible Disc FD Memorex 651 or Equiv. Drive Compatible	FD VI (Vinyl Jacket)	30712003	2.95	–	470651	–	CM-F31	FDIV	–	40002	–	FD-165	511-0	–	15026	FD65-1000	F61A111X	7910	–
Mini Flexible Disc 1s	Soft Sector (Unformatted)	3401	1.99	–	475001	54256	–	104/1	–	40500	MD1	MD 1	744-0	S/A-104	15300	MD525-01	M11A211X	7897	441002
5¼" Single-Headed Drives	10 Hard Sector	3403	1.99	–	475010	54257	–	107/1	–	40501	–	MD 110	744-10	S/A-107	15325	MD525-10	M41A211X	7898	441102
Single-Density Media	16 Hard Sector	3405	1.99	–	475016	54258	–	105/1	–	40502	MH1	MD 116	744-16	S/A-105	15326	MD525-16	M51A211X	7899	441162
	Soft Sector (Unformatted) W/Hub Ring	3431	2.19	–	–	–	–	–	–	–	–	–	–	–	–	MD525-01	–	–	–
	10 Hard Sector, w/Hub Ring	3433	2.19	–	–	–	–	–	–	–	–	–	–	–	–	MD525-10	–	–	–
	16 Hard Sector, w/Hub Ring	3435	2.19	–	–	–	–	–	–	–	–	–	–	–	–	MD525-16	–	–	–
Mini Flexible Disc 1d	Soft Sector	3417	2.24	–	–	54646	–	104/1D	–	–	–	–	–	–	–	MD540-01	–	–	–
5¼" Single-Headed Drives	10 Hard Sector	3418	2.24	–	–	54649	–	107/1D	–	–	–	–	–	–	–	MD540-10	–	–	–
Double-Density Media	16 Hard Sector	3419	2.24	–	–	54652	–	105/1D	–	–	–	–	–	–	–	MD540-16	–	–	–
Mini Flexible Disc 2d	Soft Sector	3421	2.74	–	–	54624	–	104/2D	–	–	MD2-D	–	745-0	S/A-154	–	MD550-01	–	–	–
5¼" Double-Headed Drives	10 Hard Sector	3423	2.74	–	–	54627	–	107/2D	–	–	–	–	745-10	S/A-157	–	MD550-10	–	–	–
Double Density Media	16 Hard Sector	3425	2.74	–	–	54630	–	105/2D	–	–	MH2-D	–	745-16	S/A-155	–	MD550-16	–	–	–

Memorex Flexible Discs...The Ultimate in Memory Excellence

Quality
Memorex means quality products that you can depend on. Quality control at Memorex means starting with the best materials available. Continual surveillance throughout the entire manufacturing process. The benefit of Memorex's years of experience in magnetic media production, resulting, for instance, in proprietary coating formulations. The most sophisticated testing procedures you'll find anywhere in the business.

100 Percent Error Free
Each and every Memorex Flexible Disc is certified to be 100 percent error free. *Each* track of *each* flexible disc is tested, individually, to Memorex's stringent standards of excellence. They test signal amplitude, resolution, low-pass modulation, overwrite, missing pulse error and extra pulse error. They are torque-tested, and competitively tested on drives available from almost every major drive manufacturer in the industry including drives that Memorex manufacturers. Rigid quality audits are built into every step of the manufacturing process and stringent testing result in a standard of excellence that assures you, our customer, of a quality product designed for increased data reliability and consistent top performance.

Customer-Oriented Packaging
Memorex's commitment to excellence does not stop with a quality product. They are proud of their flexible discs and they package them with pride. Both their packaging and their labeling have been designed with your ease of identification and use in mind. The desk-top box containing ten discs is convenient for filing and storage. Both box labels and jacket labels provide full information on compatibility, density, sectoring, and record length. Envelopes with multi-language care and handling instructions and color-coded removable labels are included. A write-protect feature is available to provide data security.

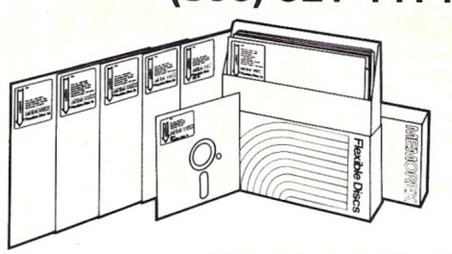
Full One Year Warranty — Your Assurance of Quality
Memorex Flexible Discs will be replaced free of charge by Memorex if they are found to be defective in materials or workmanship within one year of the date of purchase. Other than replacement, Memorex will not be responsible for any damages or losses (including consequential damages) caused by the use of Memorex Flexible Discs.

Quantity Discounts Available
Memorex Flexible Discs are packed 10 discs to a carton and 10 cartons to a case. Please order only in increments of 100 units for quantity 100 pricing. We are also willing to accommodate your smaller orders. Quantities less than 100 units are available in increments of 10 units at a 10% surcharge. **Quantity discounts** are also available. Order 500 or more discs at the same time and deduct 1%; 1,000 or more saves you 2%; 2,000 or more saves you 3%; 5,000 or more saves you 5%; 10,000 or more saves you 7%; 25,000 or more saves you 8%; 50,000 or more saves you 9% and 100,000 or more discs earns you a 10% discount off our super low quantity 100 price. Almost all Memorex Flexible Discs are immediately available from **CE**. Our warehouse facilities are equipped to help us get you the quality product you need, when you need it. If you need further assistance to find the flexible disc that's right for you, call the Memorex compatibility hotline. Dial 800-538-8080 and ask for the *flexible disc hotline* extension 0997. In California dial 800-672-3525 extension 0997.

Buy with Confidence
To get the fastest delivery from **CE** of your Memorex Flexible Discs, send or phone your order directly to our Computer Products Division. Be sure to calculate your price using the **CE** prices in this ad. Michigan residents please add 4% sales tax. Written purchase orders are accepted from approved government agencies and most well rated firms at a 10% surcharge for net 10 billing. All sales are subject to availability. All sales are final. Prices, terms and specifications are subject to change without notice. Out of stock items will be placed on backorder automatically unless **CE** is instructed differently. International orders are invited with a $20.00 surcharge for special handling in addition to shipping charges. All shipments are F.O.B. Ann Arbor, Michigan. No COD's please. Non-certified and foreign checks require bank clearance.

Mail orders to: **Communications Electronics**, Box 1002, Ann Arbor, Michigan 48106 U.S.A. Add $8.00 per case or partial-case of 100 8-inch discs or $6.00 per case of 100 5¼-inch mini-discs for U.P.S. *ground* shipping and handling in the continental U.S.A. If you have a Master Charge or Visa card, you may call anytime and place a credit card order. **Order toll-free** in the United States. Call anytime 800-521-4414. If you are outside the U.S. or in Michigan, dial 313-994-4444. **Dealer inquiries invited**. All order lines at **Communications Electronics** are staffed 24 hours.

Order Toll-Free!
(800) 521-4414



For Data Reliability—Memorex Flexible Discs

COMMUNICATIONS ELECTRONICS™ 376

Computer Products Division

854 Phoenix □ Box 1002 □ Ann Arbor, Michigan 48106 U.S.A.
Call TOLL-FREE (800) 521-4414 or outside U.S.A. (313) 994-4444

DIGITAL RESEARCH: PARTS

"TOP QUALITY PARTS FOR LESS"

Video Game Board

4.45

3 for 12.00

Hockey • Tennis • Handball

- General Instruments AY3-8500
- Features Exciting Sounds
- On Screen Scoring
- Speed & Paddle Controls
- 1 or 2 Players
- Works on 8-15 Volts D.C.

Each board comes with RF Modulator (Ch. 3 or 4) and schematic. The only parts needed to complete game are speaker, 2-1 Meg Pots & Switches.

Cable Ties

Make Your Project "Neat and Tidy." 4" Cable Ties at a Fantastic Price.

2.00 for 100
or Better Yet
1000 for 15.00

8" Heavy Duty Ties
5.00 for 100 or 40.00 for 1000

Heavy Duty Fuse Holder

1.50

New Item! Positive and Negative Leads are fused.
(Includes 2-10 Amp Fuses)

Switch Banks

- Non Canceling **2.50**
- DPDT-PC or Solder
- Switches Easily Removed
- Push On/Push Off

THAT'S INCREDIBLE!

Gold Wire Wrap Sockets

Not Gold Inlay as Sold By Others.
Super 3 Level Gold Wire Wrap.

14 Pin - **10/3.95, 25/8.75**
16 Pin - **10/4.95, 25/11.25**

Infared Photocell (Dual)

69¢

Peaks at 800 Nano Meters
Dark Resistance
Dark › 50 Meg • Light ‹ 1K

Video Paddle Controls

2 for 1.00

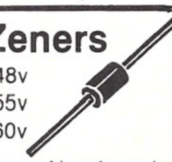
1 Meg
Can be used with game board above.

Power Transistor TO220 Case

3/1.10

1 Amp 30 Watts 100 Volt
TIP 30C (PNP)
TIP 29C (NPN)

5 Watt Zeners



6.8v	15v	48v
11v	24v	55v
12.6 v	30v	60v

5% • Some House Numbered
Cut & Formed • Prime
10 for 1.00

IN4735 • 6.2V • 1 Watt
Cut & Formed **20/1.00**

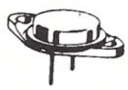
IC Specials!

LM1889 - **2.25**
MC1310 - **1.80**

TWO GREAT BUYS!

Voltage Regulator

LM309K



1.00

5 Volt - 1 Amp Regulator
TO3 Case. Super Special!

Sprague RFI Filter

3.65 or 3/9.00

Perfect for Computers, or anything that needs to be "glitch" free. By the #1 name in filtering, Sprague. JN17-5109B. Has I.E.C. Power Line Connector. 2x3 Amp. 115/220 VAC 60 Hz. 2⅛" x 2½" x 3" deep.

More IC's

Repeat of a Sellout
4558-Dual 741 in 8 pin Mini Dip.

4/1.20

NE555 Timer
3/1.25 or 10/3.30

Small Torroid

O.D. = ½"
I.D. = 5/16"

6/1.00

Filter Choke
1.5 MH Molded
8 for 1.00

Rectifier Diode

IN4007

11/1.00

1000 Volts, 1 Amp
DO-41 Case • Prime •
Long Lead • Marked.

RCA Triac

79¢

5 for 3.50

T2800M-TO220 Case
6 Amp 600 Volt

JFET OP AMP

Super High Input Impedance (10^{12} OHMS) — High Frequency Response. TO 4 MHZ. Large DC Voltage Gain 106 DB — New generation OP-AMP with Vastly Superior Features!

LF356BH - **75¢ or 3/2.00**

Micro Mini Toggle Switch

99¢

6 for 5.00

SPDT • Made in USA with Hardware

DON'T MISS THE "ELECTRONIC IGNITION BOARD" ON PAGE 5 OF OUR EXCITING NEW CATALOG!

TERMS:
Add 50¢ postage, we pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa, MasterCard and American Express cards. Tex. Res. add 5% Tax. Foreign orders (Canada 10%) add 20% P & H.

• VISA • MASTERCARD • AMERICAN EXPRESS •

Digital Research: Parts
P.O. Box 401247 • Garland, Texas 75040
(214) 271-2461

kilobaud

MICROCOMPUTING T.M. LIST OF ADVERTISERS

For further information from our advertisers, please use the Reader Service card.

*This advertiser prefers to be contacted directly.

cution space, speed or efficiency. This sequence uses the assembler as the subtracter: load track number, add address of DDATA – 1 fetch from memory.

I have found that the preferred numbering scheme is usually clear from the application and the language facilities. What is probably most important is that the program comments and/or documentation makes it clear as to which scheme is used, and why.

Richard A. Holmes
Las Vegas, NV

Polyalgorithms

Bob Fruit's letter in the April issue (p. 29) concerning Arnold Bragg's "Area Estimation" prompts this one.

Bragg's algorithm is indeed simple, but not as simple as it could be. Listing 1 shows a BASIC rendering of an even simpler implementation. The basic idea is that the determinant of a matrix is the signed area of a certain parallelepiped. If, for example, the matrix is one whose first row is (2, – 1) and second is (1,1), then its determinant, 3, is the area of the parallelogram whose vertices are (0,0), (2, – 1), (1,1) and (3,0) (= the sum of (2, – 1) and (1,1)). The ordering of the rows of the matrix determines the sign. Half this area is the area of the triangle with vertices (0,0), (2, – 1) and (1,1). Summing these signed areas over an appropriate sequence of vertex pairs gives the signed area of an arbitrary polygon (unwanted area cancelling out nicely).

One such polygon is the one Fruit mentions whose vertices are on a ring. Hence Bragg's algorithm still works in one pass for such a figure. For a simple example consider the "square" ring whose vertices are (1,1), (1, – 1), (– 1, – 1), (– 1,1), (2,2), (2, – 2), (– 2, – 2) and (– 2,2). Use Bragg's algorithm traversing the points in the order (2,2), (2, – 2), (– 2, – 2), (– 2,2), (2,2), (1,1), (– 1,1), (– 1, – 1), (1, – 1), (1,1), i.e., clockwise around the outside and counterclockwise around the inside.

William A. McWorter, Jr.
Ohio State University
Columbus, OH

H-89 Software

Your review of the H-89 ("Building the H-89," March–April 1981) mentions software availability as an area of concern. In this regard, your readers should know about The Software Toolworks.

We are suppliers of software mainly for the Heath H-89/Z89 and H8, although some of our products are also available in eight-inch CP/M format. Our software is carried by almost all the Heathkit Electronics Centers in the United States and Canada.

The H-89 display and function keys are perfectly suited to a full screen file editor, and we offer one which is exceptionally easy to learn and use. We also have a text formatter which combines with the editor to provide economical and versatile word processing capability and a full screen graphics editor.

Our product line contains many languages: a C compiler, RATFOR preprocessor for Microsoft FORTRAN, LISP interpreter, and Z-80 and 8080 absolute macro assemblers. Other offerings include a modem and file transfer program, a disk catalog and utilities package, and file compression and encryption.

Entertainment programs available are an Invaders game, the championship level Mychess program and an air traffic controller game, all with full graphics for the H-89.

All our software is in the $20 to $40

```
10 '          *** PROGRAM FOR THE SIGNED AREA OF A POLYGON ***
20 '
30 PRINT"ENTER VERTICES IN THE ORDER A BORDER GUARD WOULD TRAVERSE THEM"
40 PRINT"KEEPING THE INTERIOR OF THE POLYGON TO HIS RIGHT. THE ENTRY"
50 PRINT"SYNTAX IS X,Y,(2ND COMMA REQUIRED). SIGNAL THE END OF ENTRIES"
60 PRINT"BY ENTERING A THIRD NON-ZERO NUMBER WITH THE LAST ENTRY."
70 PRINT"(CHANGING THE ORDER AND/OR THE LAST ENTRY EXTENDS THE FLEXIBILITY"
80 PRINT"OF THIS PROGRAM TO COMPUTE INTEGRALS OF FUNCTIONS AND THEIR"
90 PRINT"ABSOLUTE VALUES)"
100 '
110 '          COMPUTE 2*AREA AS VERTICES ARE INPUT
120 '
130 INPUTX,Y,Z:A=A+X*T-S*Y:PRINTA/2:IFZ=0THENS=X:T=Y:GOTO130
140 '
150 '          PRINT RESULT
160 '
170 PRINT:PRINT"THE SIGNED AREA IS":A/2
180 '
190 '          RERUN THE PROGRAM
200 '
210 PRINT:A=0:S=0:T=0:Z=0:GOTO130
```

Listing 1.

price range. Nonetheless, it works and works well.

Walt Bilofsky, Prop.
The Software Toolworks
14478 Glorietta Drive
Sherman Oaks, CA 91423

Any Interest?

We would like to hear from anyone interested in subscribing to a network of hobby computers in which the central system would place calls to all other users during the evening hours (reduced rates).

The system would be similar to many of the "bulletin board" systems now available, but messages would be delivered and picked up by the central system instead of the users having to call in.

We don't know if such a system will be feasible, but we would appreciate it if anyone interested in investigating the possibilities would send us any comments, questions or suggestions, along with a stamped, self-addressed envelope.

We would also be interested in corresponding with people interested in computer-generated music, particularly involving the Mountain Computer Music System for the Apple.

Donna and Michael Witt
PO Box 55686
Valancia, CA 91355

TRS-80 Compiler

I am attempting to compile a list of TRS-80 Model I programs that will (or will not) run on the Model III. I would appreciate any input from your readers.

To those who kindly respond, please mention if the program was on disk or tape—BASIC, SYSTEM or disk CMD file—whether run under Level II or DOS—and, if DOS, which one. If any changes are needed to be made to the program, what were they?

All those who send me information will be sent the compiled list (after a reasonable length of time to get all input) *if* an SASE is included with your information.

Ken Knecht
1340 W. 3rd St. #130
Yuma, AZ 85364

The Games People Play

I very much agree with your editorial on games ("Short-Lived Computer Games," April 1981, p. 6) and how they become boring after awhile. That is because there are no games with extended play value. I think they are improving, however; Super-Invader, ABM and Galaxian, as well as the Adventures, are entertaining. But there is a deadly sameness to the computer as an opponent, and even when you engage in classic games such as chess and backgammon, it is not the same as playing against a live, variable opponent.

But that doesn't mean computers can't ever play games. There is still the pragmatic position of "guess what 100 kids surveyed reached for when they had a choice between a board game and a computer?"

There has to be an area in computing for the game. It is just that it hasn't developed and focused yet.

Allan Turoff
Snedens Landing, NY

Making It Work

I particularly enjoyed the article "Microcomputer Selection and Implementation" by Bruce T. Pace (March 1981, p. 120). It would be nice to see more articles like it which highlight the "hidden" factors in a successful data processing installation.

The average person has been led to believe that the computer is some mysterious black box that either works or doesn't. The fact is, it takes real effort and planning to make it work! This applies to large computer installations as well as small businesses contemplating their first machine, which may very well be a micro.

The temptation is there for a vendor to oversell and highlight the ease with which the computer will fit the customers' needs. And thankfully, usually it does. However, Mr. Pace makes a strong argument in favor of a vendor-independent consultant. When the neophyte computerist, or even the person who has only used a micro for games, goes out to acquire a business system, the requirements may not be altogether obvious. Who thinks of a contingency plan? Or, one point that almost got missed, can the system be expanded when the business grows? Thinking about and caring for these very important matters at the outset can save much heartache later.

All things considered, I thought Mr. Pace's article was quite thorough and fair. Keep up the good work!

Daryl A. Frame
President
Discreet Computer Service, Inc.
Ventura, CA 93003

Sorting Things Out

In "Sorta Super Fast" by Albert J. Marino (April 1981, p. 164), the algorithm in Fig. 1 which Mr. Marino has "never seen . . . implemented and might be unique," is Algorithm 175, Shuttlesort, by C. J. Shaw and T. N. Trimble of System Development Corp. (Santa Monica, CA), which appeared in the *Communications of the Association for Computing Machinery* in June 1963. The method is also known as sifting.

In the same issue, "A Better Bubble Sort" by W. A. Harrison and S. H. Sachs (p. 148) contains several errors in the flowcharts. First, the exchange takes too many steps and one too many temporary variables. Only one "temporary" is needed for the exchange:

$x = key_j$
$key_j = key_{j+1}$
$key_{j+1} = x$

If the point seems minor, consider that this code is executed often, and the records might be several bytes long.

The sort is not "stable"; i.e., identical keys will be swapped, destroying any previous ordering within the list, which may have been imposed by an earlier sort on another field in the case where the sort-key is not the entire record, or perhaps the file was created chronologically as events occurred to produce the records. If the sort-key is the entire record, the sorted list will look the same, so no apparent harm is done; however, the swap is unnecessary and wastes time. In the bubble sort, this may be the only swap during a pass, and will indicate the need for another pass, which is totally unnecessary. The swap should be done only if a key is greater than its linear successor in the list, to make this a stable sort.

Both the one-way and two-way algorithms fail to recognize that each pass "ranks" one key by migrating it to its final position in the list. The ranked key need not participate in subsequent passes; each pass handles a smaller unsorted list. For the one-way sort, the sorted list "grows" downward from the largest key, leaving the unsorted list at the beginning of the list; for the two-way, another list is growing upward, leaving the unsorted list in the middle. By failing to eliminate ranked keys, the "better" bubble may perform *twice* as many comparisons as a properly coded one-way sort.

Since no keys are eliminated from the unsorted list, and identical keys are swapped, and any swap requires another pass, the occurrence of identical keys will prevent the sort from terminating—an infinite loop. Eventually, the list would be sorted, but the identical keys would be the only swap performed, triggering another pass to do the same thing—forever.

The Bubble and Better Bubble sorts are by no means efficient. The Bubble is one of a number of sorting methods which are sensitive to the initial disorder of the list; in this case, positive displacement (a key occurring later in the list than its rank) is a big problem. The Better Bubble minimizes this sensitivity by creating an equivalent sensitivity in the other direction which helps only in some cases. For a reversed list, or any case where the sum of the positive and negative displacements is zero, the Better Bubble performs no better than the one-way Bubble.

P. V. Piescik
Wethersfield, CT

MODEL II

26-4002
64K 1 Drive
$3440.00

26-4160 1 Drive Exp........$1034.00
26-4161 2 Drive Exp.........1574.00
26-4162 3 Drive Exp.........2114.00
26-4530 Scripsit II............265.00
26-4512 Profile II............162.00
26-4511 Visicalc II...........265.00
26-4501 Gen Ledger.........180.00
26-4506 Mail List.............72.00

$ DISCOUNT $

TRS-80®

DEALER A301

COMPUTER SPECIALISTS

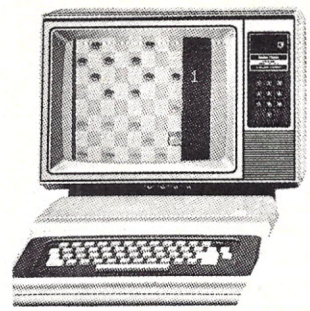
CALL US. . .
SAVE MONEY

We carry the full line of TRS-80 Computers. All other R.S. software, furniture, and accessories at discount from catalog price. We stock most items to assure you fast delivery and save you money.

MODEL III

26-1061 4K I.............$629.00
26-1062 16K III...........865.00
26-1063 32K III
W/2 Drives, RS232.......2225.00

COLOR



26-3001 4K..............$353.00
26-3002 16K Ext. Basic.....533.00
26-3008 Joysticks..........22.50
26-3010 Color Video......353.00
26-1206 Recorder..........54.00

PRINTERS

26-1140 Expansion Interface......................$249.00
26-1141 16K Exp. Interface.........................359.00
26-1142 32K Exp. INterface.......................469.00
26-1145 RS232C Board.............................84.00
26-1160/1 Mini Disk Drive.........................419.00
26-1563 Scripsit-Disk..............................79.00
26-1566 Visicalc..................................83.00

* * * * * ***PRINTERS*** * * * * *

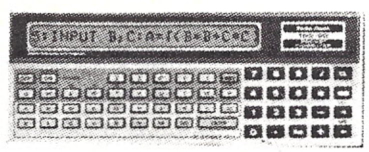
26-1155 Quick Printer............................187.00
26-1167 9½ Dot Matrix Printer.....................360.00
26-1166 Line Printer VI..........................1080.00
26-1158 Daisy Wheel II..........................1799.00
26-1165 Line Printer V............................1710.00

CENTRONICS

BEST PRICES

Fast 100 CPS Centronics
730-1A Printer.............$577.00
Text Quality Centronics
737-1 Printer...............$737.00

Pocket Computer



26-3501 1.9K P.C...........$221.00
26-3503 Cassette IF...........45.00
14-812 Recorder..............72.00

commodore
AUTHORIZED DEALER
CALL FOR PRICES

ALL POCKET AND COLOR COMPUTER SOFTWARE SOLD AT DISCOUNT

WRITE US FOR A
FREE CATALOG

1-800-841-0860 Toll Free Order Entry

MICRO MANAGEMENT SYSTEMS, INC. ✓100

No Taxes on Out Of State Shipments

Immediate Shipment From Stock on Most Items

DOWNTOWN PLAZA SHOPPING CENTER
115 C. SECOND AVE. S.W.
CAIRO, GEORGIA 31728
(912) 377-7120 Ga. Phone No. & Export

TRS-80 is a registered trademark of the Tandy Corp.

R.S. 90 Day Limited Warranty
F-48 Form Provided

Largest Inventory
In The S.E. U.S.A.

✓See List of Advertisers on page 210

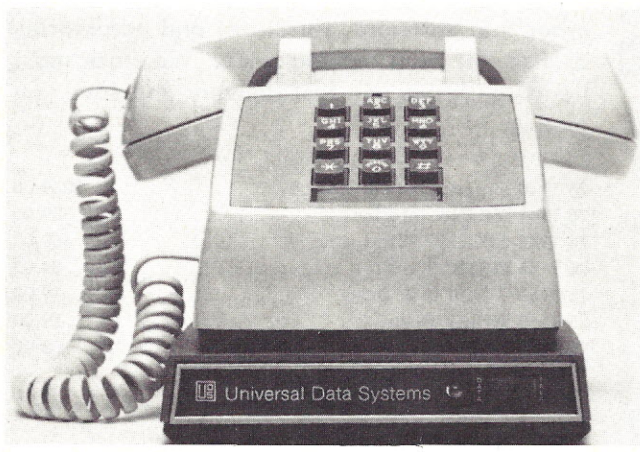
Edited by Linda Stephenson

64K RAM for S-100 Bus
Automatic Answer Modem
Power Line Monitors
Talk to Your Micro

64K Static RAM for S-100 Bus

CompuPro Division of Godbout Electronics, PO Box 2355, Oakland Airport, CA 94614, is now offering the RAM 17—a 64K fully static memory board for S-100 IEEE 696 computers. This static board features faster access time and lower power dissipation (less than 2 W) than its dynamic counterparts. It offers freedom from alpha particle soft-bit errors and problem-free DMA operation.

The RAM 17 is available as partially assembled "unkit," assembled and tested or certified system component. Prices are $1095, $1395 and $1595, respectively. Reader service number 466.



UDS auto-answer modem.

Automatic Answer Modem

A direct connect modem, capable of full duplex operation at 300 bits per second, and providing automatic answer service, is available from Universal Data Systems, 5000 Bradford Drive, Huntsville, AL 35805. A manual switch on the 103J LP backpanel selects either originate or answer frequency pairs. The modem will automatically answer incoming data calls in the answer mode when the front panel selector is set to data, regardless of how the switch is set. It's slightly more than one inch high. Connections are via RJ-11-type snap-in modular plugs. The unit has an RS-232C digital interface. Price is $245. Reader service number 476.

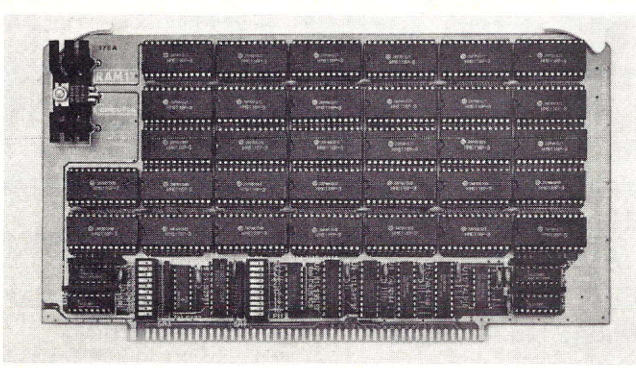
Line Monitor Power Conditioners

Power Master line monitor power conditioners reduce electrical pollution coming through electrical power lines and branch circuits to solid state electronic equipment that is sensitive to voltage spikes and electrical noise. The product line consists of eight power conditioner models which vary in complexity from a wall plug-in unit to a sophisticated console or rack-mounted unit. The simplest models contain varistors to clamp or reduce voltage spikes. The more sophisticated units include a multistage filter network with spike suppression capability, transverse and common mode electrical noise filtration, and an undervoltage warning system, with lights that indicate when the incoming electrical line voltage drops below a safe level. An ac voltmeter allows the operator to inspect incoming ac power.

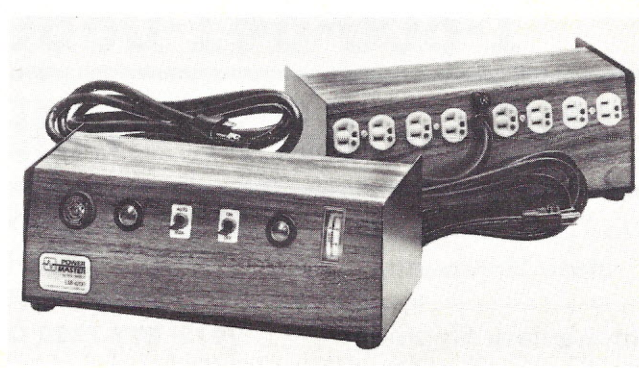
SGL Waber Electric, 300 Harvard Ave., Westville, NJ 08093. Reader service number 483.

Speech Synthesizer

Type-'n-Talk text-to-speech synthesizer allows a hobby-



RAM 17 static memory board.



Power Master line monitor power conditioners.

AUTOMATED EQUIPMENT INCORPORATED

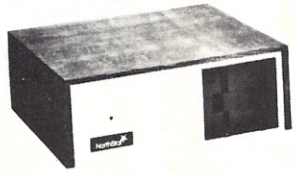
↙96

18430 Ward, Fountain Valley, CA 92708

Call Toll-Free for latest low prices!
1-800-854-7635 Outside CA
714-963-1414 Inside CA
1-800-854-7635 Outside Cont. USA
Prices change daily to meet competition.

TERMINALS

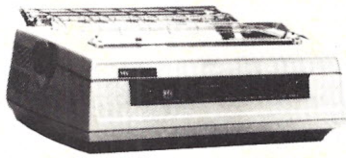
TELEVIDEO 912 B	CALL
TELEVIDEO 912 C	CALL
TELEVIDEO 920 B	CALL
TELEVIDEO 920 C	CALL
SOROC IQ-120	675
ZENITH A-19	750
HAZELTINE 1500	995
TELEVIDEO 950	CALL



NORTHSTAR

HRZ-1D-32K	CALL
HRZ-2D-32K	CALL
HRZ-1Q-32K	CALL
HRZ-2Q-32K	CALL
ADDITIONAL 16K RAM	CALL
ADDITIONAL 32K RAM	CALL
HARD DISK SYSTEM	CALL

SPECIAL MEMORY CONFIGURATIONS AVAILABLE, INCLUDING SYSTEMS GROUP.



PRINTERS

NEC 5510 (TRACT., RIB., THIM)	2600
NEC 5515 (TRACT., RIB., THIM)	2600
NEC 5520 (TRACT., RIB., THIM)	2900
NEC 5530 (TRACT., RIB., THIM)	2540
TI-810	1495
TI-820	1625
EPSON MX 80	CALL
EPSON MX 70	CALL
EPSON MX 100	CALL
ANADEX DP-9500/9501	CALL

VIP's customarily need top quality and service within stringent budgetary limits. naturally, they call A.E.I.

Item: A.E.I. tests virtually every* item sold, before shipping.

Item: A.E.I. initializes every piece of software, and makes a copy. If you have a problem, A.E.I. can check to locate the problem by using the in-house duplicate, saving you time.

Item: A.E.I. has in-house expetise on every item offered.

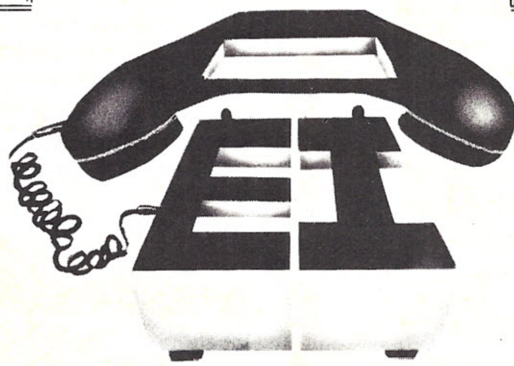
Item: A.E.I. prices are competitive with the lowest, no-service, no in-house expertise dealer.

40% OF ALL A.E.I. SALES ARE TO PUBLIC AND SEMI-PUBLIC INSTITUTIONS.

A partial list** of A.E.I. customers who purchased during the last quarter of 1980 includes: University of Nebraska, University of Virginia, U.S. Dept. of Interior, University of Kentucky, University of California, Massachusetts Institute of Technology, U.S. Air Force, Naval Air Development Center, Brown University, University of Oregon, University of Utah, Ohio State University, University of Southern California, California State Dept. of Water Resources, University of Michigan, and Princeton University.

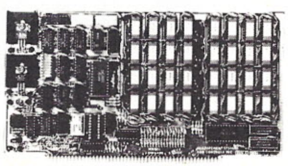
In the private sector, an equal number of major national organizations purchased during the same period.

You can call A.E.I. too!



VIP's Call A.E.I.

*With our normal fast shipping, we have time to test; expedited shipment may preclude testing.
**A.E.I. does not wish to imply that any of these fine organizations endorse A.E.I., merely that A.E.I. is proud to have them as customers.



S.S.M.	KIT	ASM
CB2 Z80 CPU	$200	$265
VB1C VIDEO	140	190
VB2 VIDEO	155	210
VB3 VIDEO	375	440
IO4 INTERFACE	165	225
SB1 SYNTHESIZER	195	270

MEASUREMENT SYSTEMS MEMORY

DM3200 32K 4MHZ	480
DM6400 64K 4MHZ	595
DMB3200 32K 4MHZ BANK SELECT	630
DMB6400 64K 4MHZ BANK SELECT	745

DYNABYTE SYSTEMS ALSO AVAILABLE—CALL FOR THE EXCITING NEWS ABOUT NEW COMPETITIVE PRICING!

WE WILL TRY TO BEAT ANY ADVERTISED PRICE.

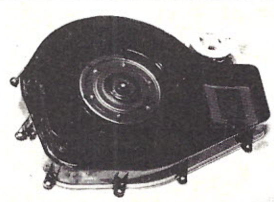
TERMS: All prices listed are cash discounted and subject to change. COD's are 2% additional. Universities and well-rated firms Net-10.

SHIPPING: ADDITIONAL IN ALL CASES.

TECHNICAL ASSISTANCE:
WE TRY TO HELP INTERFACE AND TROUBLESHOOT—CALL (714) 963-1414.

Open Monday-Friday, 8:00 a.m.-5:00 p.m.

Dealer Inquiries Invited.



MORROW

DECISION 1 BASIC	$1550
DISCUS-1 2 DRIVE	1550
DISCUS 2D 1 DRIVE	935
DISCUS 2D 2 DRIVE	1550
DISCUS 2+2 1 DRIVE	1250
DISCUS 2+2 2 DRIVE	2190
DISC JOCKEY 2D	350
DISCUS M26 HARD DISC	3925
ADDITIONAL HARD DISC	3685
DISCUS M10	2990

MISCELLANEOUS

NEC THIMBLES	$16
NEC RIBBONS	7
RS2332 CABLE 5′	20
RS 232 CABLE 10″	25
LEDEX MONITOR	145
NOVATION CAT	145

SOFTWARE

WORDSTAR	$330
MAILMERGE	100
DATASTAR	250
N.S. PASCAL	160
MAGIC WAND	290
C BASIC	100

GRAHAM-DORIAN

JOB COSTING	$700
INVENTORY	475
CASH REG.	475
APARTMENT	475
MEDICAL	700

ADDITIONAL SOFTWARE DISCOUNTS WITH SYSTEM PURCHASE

DISCS—BOX OF 10

VERBATUM 5¼ 1 SIDE	$27
VERBATUM 5¼ 2 SIDE	45
VERBATUM 8 1 SIDE	35
VERBATUM 8 2 SIDE	55
OTHERS	CALL
PLASTIC STORAGE BOXES	3

OUR SYSTEMS WORK!

All systems normally tested and configured in our repair facility before delivery. Service contracts available. Prompt repairs and warranties.

Type-'n-Talk by Votrax.

ist's personal computer to talk back to him in intelligible English words and phrases. Used with any computer that has an RS-232C interface, Type-'n-Talk lets you type an unlimited combination of English words and phrases on the keyboard. The computer will then "speak" the words as they are typed, or they can be held in the 750-character buffer until the user prompts the computer to speak them in entire phrases or sentences. The unit costs $345.

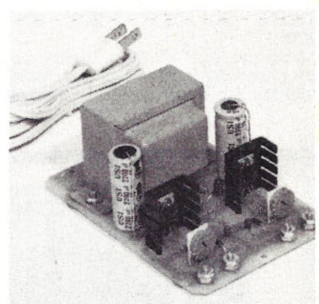
Votrax, 500 Stephenson Highway, Troy, MI 48084. Reader service number 485.

Electronic Typewriter Interface

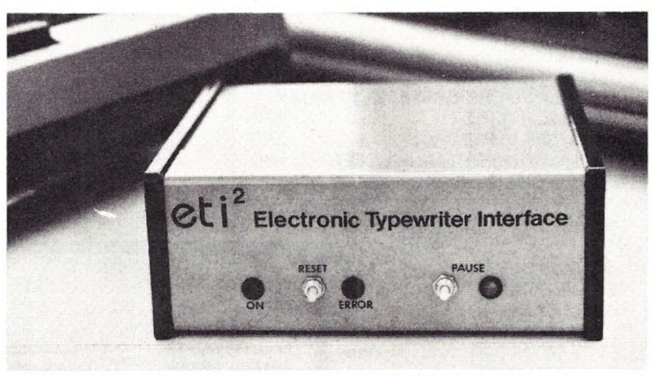
The ETI² interface from Mediamix, PO Box 67B57, Los Angeles, CA 90067, makes it possible to connect an IBM electronic typewriter model 50, 60 or 75 to any computer having a standard parallel printer port. The ETI² features a Z-80 microprocessor, 2K of RAM (text buffer) and total access to all of the typewriter's automatic functions (word and phrase underlining, tabbing, centering, etc.). The user can easily redefine the ASCII to IBM character and control codes translation. This feature allows most word processing programs to access the typewriter's automatic functions, without modifying the word processing program. Price is $495. Reader service number 467.

Adjustable Dual Power Supply Kit

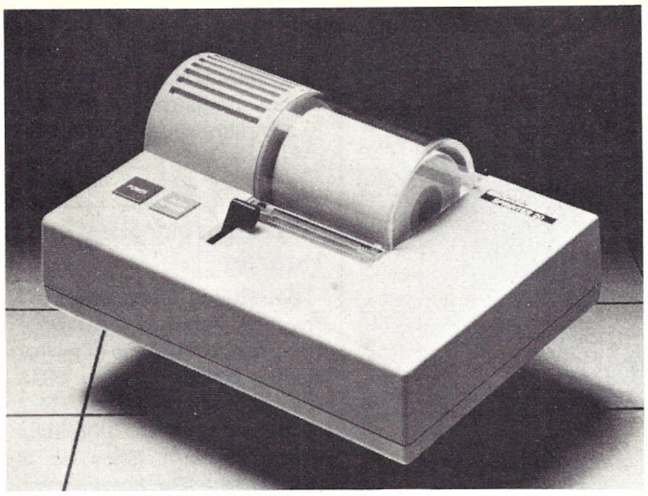
Recently announced by Jameco Electronics, 1355 Shoreway Road, Belmont, CA 94002, is the model JE215 ad-



JE215 adjustable dual power supply.



The ETI² interface connects an IBM electronic typewriter with any computer's parallel printer port.



Sprinter 20 printer/plotter.

justable dual power supply kit. The supply provides independent adjustable positive and negative output from 5 to 15 V dc regulated. Separate adjustment for each supply allows the user unlimited applications for integrated circuit requirements. Power output for each supply ranges from 5 V dc at 500 mA to 15 V dc at 175 mA, and an on-board LED indicates power-on condition. Price is $24.95. Reader service number 484.

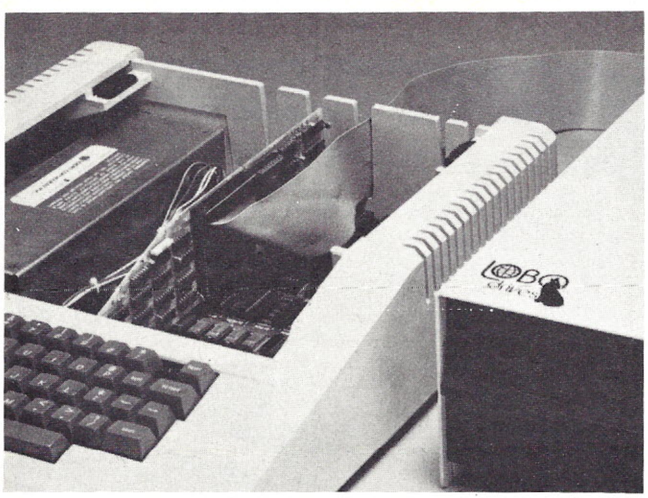
20-Character Matrix Printer/Plotter

Sprinter 20, a high-speed matrix printer/plotter, provides program control of print mode. The Sprinter 20 prints a full 280-line CRT display in 28 seconds. The user can select parallel seven-bit ASCII or serial RS-232 with selectable baud rates from 110 to 9600. The print mode is 20 5 × 7 dot matrix characters per line with 96 ASCII upper and lowercase characters. Programmable ASCII controls include automatic carriage return and line feed, right justification, form feed, graphic control, multiline feed and strip printing. Price is $175.

Alphacom, Inc., 3031 Tisch Way, San Jose, CA 95128. Reader service number 481.

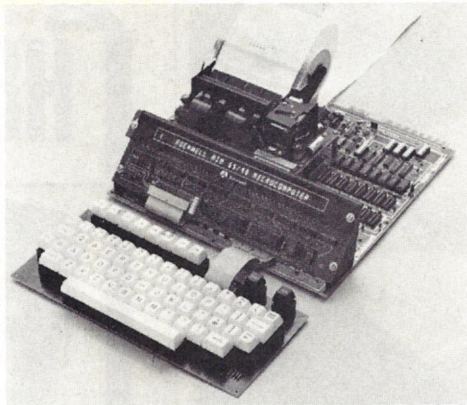
Floppy Disk Controller for Apple II

Lobo Drives International, 354 South Fairview Ave., Goleta, CA 93117, has a new disk controller board for the Apple computer. The LCA-22 is software compatible with Apple DOS and contains 256 bytes of on-board boot ROM. It



Lobo disk controller board.

The Microconnection direct connect modem.



Rockwell AIM 65/40 microcomputer.

will control up to four eight-inch single- or double-sided, single- or double-density disk drives with a total storage capacity of 4.4 megabytes. Controller card, cable, diskette and documentation are $699. Reader service number 482.

Direct Connect Modem for the Atari

The Microconnection for the Atari is a direct connect modem which eliminates the need for acoustic coupled modems. The modem is connected to the computer console data port and employs software for 300 baud data I/O. It can be used for accessing The Source, Micronet and computer bulletin boards, and for data transfer with other computers.

The Microconnection features a serial printer interface. Any printer capable of 300 baud will simultaneously reproduce whatever appears on the screen. A cassette recorder can be plugged into the Microconnection and will spool on-line communications for later playback. Price is $249. Similar Microconnections are available for recently introduced TRS-80 models.

Microperipheral Corporation, 2643 151st Place, N.E., Redmond, WA 98052. Reader service number 475.

Interface Module for System-50 Computers

Percom Data Company, 211 N. Kirby, Garland, TX 75042, is offering a dual serial I/O module for System-50 computers. System-50 computers are developed around 680X microprocessors. The Percom dual-channel I/O card is designed for the 30-pin I/O bus, the SS-30 bus of System-50 computers. The module works with both older and newer versions of the SS-30 bus, and provides for an optional on-card bit rate generator for applications where a system baud clock generator is not available. I/O card with bit rate generator is $74.95. Reader service number 478.

Modular Microcomputer

The AIM 65/40 professional microcomputer comprises four modules—an R6502-based single board computer with on-board expansion to 65K, a printer with full graphic 280 × n dot matrix and 40-column alphanumeric modes, a 40-character alphanumeric display and a full ASCII keyboard with user-assignable function keys. The AIM 65/40 incorporates a ROM-resident software system which integrates all four modules. The interactive monitor software controls the system with single-keystroke, self-prompting commands, and supports software development with assembler, debug and control commands. A multifile text editor supports both line- and screen-editing functions. Price is $1795.

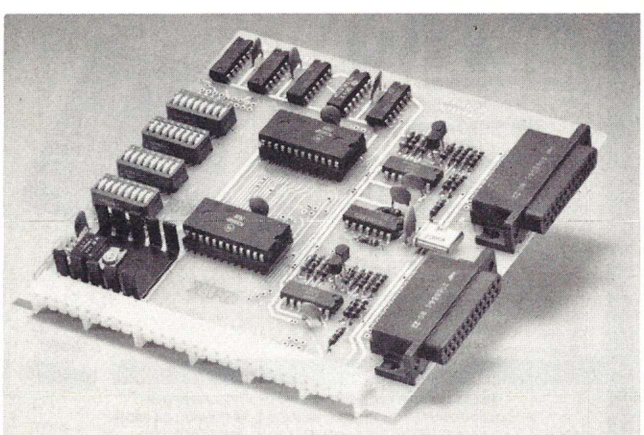
Rockwell International, Electronic Devices Division, 3310 Miraloma Ave., PO Box 3669, Anaheim, CA 92803. Reader service number 473.

STD Prototyping Boards

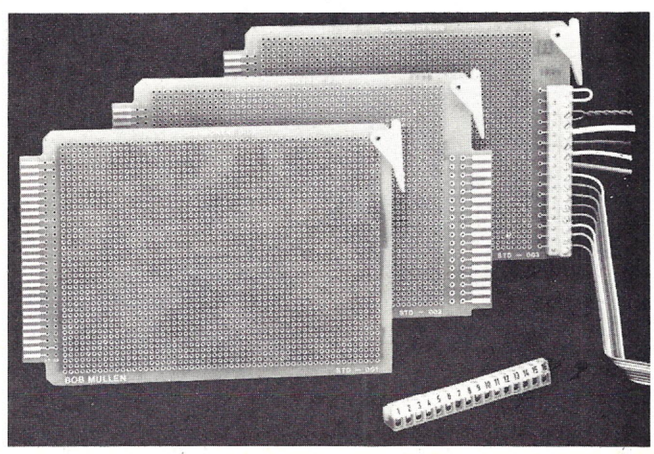
Bob Mullen, 2306 American Ave., #6, Hayward, CA 94545, has introduced a trio of prototyping boards for the STD bus. Each is .0625-inch FR-4 glass epoxy with .038-inch plated through holes on a .1-inch grid. The boards have solder plate over one-ounce copper and have 30 microinch gold on connector surfaces. The STD-001 has a 56 × 41 grid and is spaced to allow connectors for flat cable up to 50 conductors. The STD-002 has a 53 × 41 grid and a 36-position edge connector. The STD-003 has a 51 × 41 grid, equipped with a 16-position terminal strip capable of handling 15 A and 300 V. Prices: STD-001, $29; STD-002, $34; STD-003, $39. Reader service number 468.

Three New Disk Controllers

Gimix, Inc., 1337 West 37th Place, Chicago, IL 60609, is offering three new floppy-disk controller boards for SS-50 bus, 6809 and 6800 systems. The DMA controller uses high-speed direct memory access for data transfers to and from system memory. It supports any combination of drives: eight- and five-inch, single- and double-track, single- and double-density—up



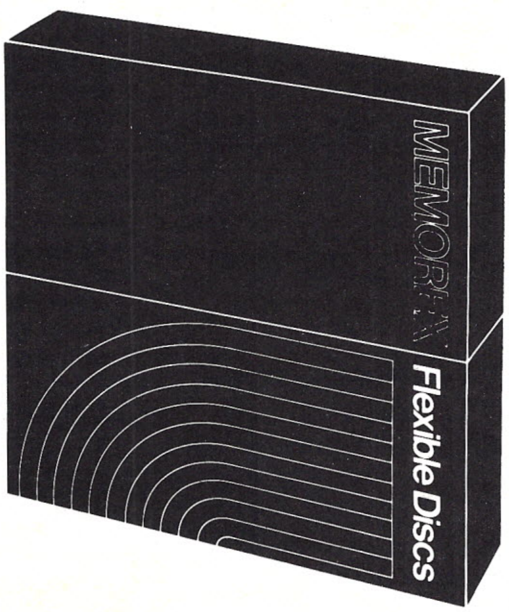
Percom's interface module for System-50s.



Prototyping boards for the STD bus.

MEMOREX

WHEN YOU ADD IT UP, MEMOREX FLEXIBLE DISCS FROM PACIFIC EXCHANGES ARE YOUR BEST BUY.

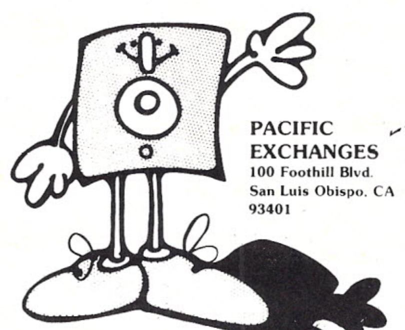


For Data Reliability—
Memorex
Flexible
Discs

CALL FREE FOR PRICES & INFORMATION

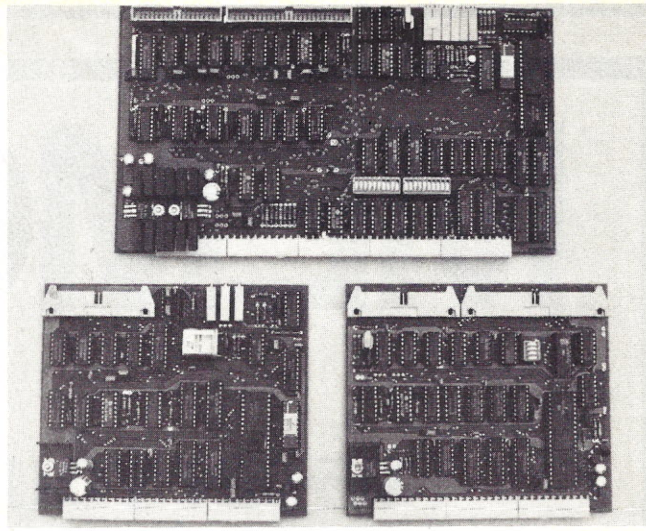
(800)235-4137

In California, (805)543-1037



PACIFIC EXCHANGES ✓172
100 Foothill Blvd.
San Luis Obispo, CA
93401

DEALER INQUIRIES INVITED VISA master charge C.O.D.'s ACCEPTED



Gimix floppy-disk controller boards.

to four drives total. The board features both a phase-locked loop data separator and adjustable write precompensation to ensure reliability, and can be used in 6809 systems running at 1, 1.5 and 2 MHz. Price is $548.68.

The double-density PIO controller (6809 only) can be used at 1, 1.5 and 2 MHz in systems that have slow I/O circuitry on the motherboard. Price is $348.28.

The Gimix 5/8 controller can be used in both 6809- and 6800-based systems. In 6800 systems the board can only be used with five-inch drives. Priced at $226.58. Reader service number 470.

6809 Single Board Computer

The ADS 6809 S-100 single board computer brings the capabilities of the 6809 to the S-100 bus. The board features provisions for 2K of RAM, 4K/8K or 16K of EPROM, RS-232 serial communications with switch-selectable baud rates, parallel I/O ports using a PIA and simulated 8080-type I/O. A 2K monitor, ADSMON, is available to quickly get you up and running. ADSMON lets you examine and change memory and register, test memory, calculate relative offsets and load and punch tape files. Assembled and tested with 2K RAM and ADSMON, the price is $449.50.

Ackerman Digital Systems, 110 No. York Road, Suite 208, Elmhurst, IL 60126. Reader service number 469.

Intelligent Interface for Computer Control of Video Disks

SSM Microcomputer Products, 2190 Paragon Drive, San Jose, CA 95131, is offering a 6802-based interface unit that allows read-only video disks to be controlled from a computer keyboard. The Universal External Interface was designed and built by SSM for DiscoVision Associates, Costa Mesa, CA, which markets laser-based video disk playback equipment.

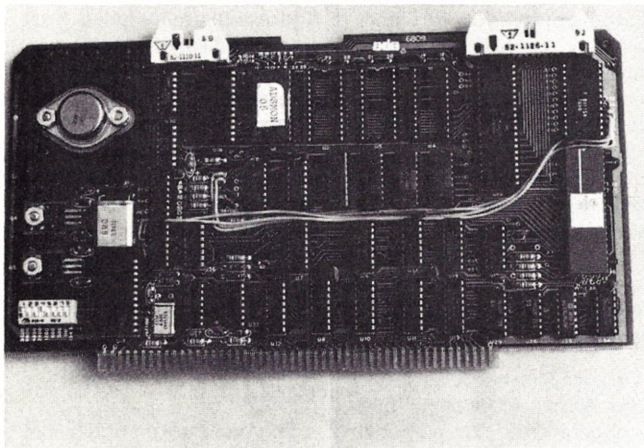
The computer operator can interrogate and directly access any portion of the recorded information; average access time is about one second. The UEI is available for either serial RS-232 use or IEEE 488 bus-based systems. Reader service number 472.

Tiny BASIC Computer Module

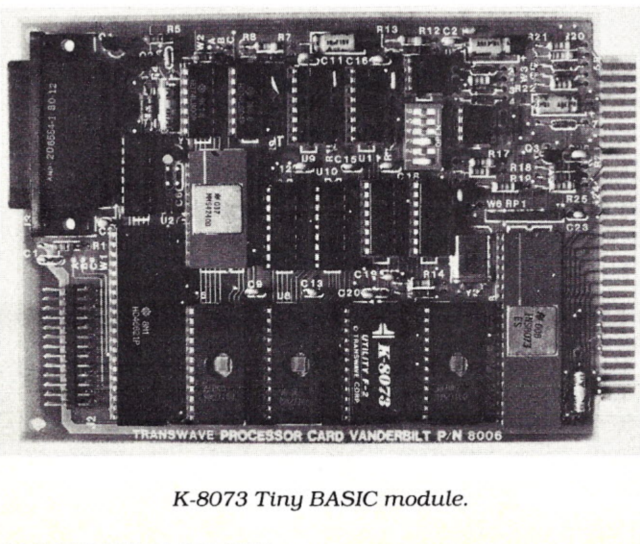
The K-8073 board from Transwave Corporation, R.D. 1, Box 489, Vanderbilt, PA 15486, provides varied design applications using National's INS8073 single-chip Tiny BASIC microinterpreter MPU. The K-8073 can be used as a development system or as a stand-alone dedicated computer. Features include serial communications, select-

SSM's UEI puts video disks under control of a computer keyboard.

able from 110–4800 baud (RS-232), cassette tape in/out, 8K of plug-in EPROM for program storage with automatic execution, on-board programmable EPROM slot and 1K of local RAM. The computer comes with 2K ROM firmware. Supplied as a 4.5 × 6.5-inch card, or modular housed, for $388. Reader service number 479.



ADS 6809 single board computer.



K-8073 Tiny BASIC module.

TRS-80 RTTY System
Apple II Financial Planner
PL/1-80 Programming Language
Cheaptalk for the TRS-80

TRS-80 RTTY System

Mailbox is a disk-based RTTY system for the TRS-80 Models I and III. This program transforms a TRS-80 disk system with at least 32K RAM into an RTTY communications terminal. It features disk-based WRU with mailbox storage and retrieval of messages, programmable messages, and storage and replay of received text. Mailbox permits transmission and reception of BASIC and assembly source listings, object code and data files in Intel hex format with error checking. It communicates through the Macrotronics Ham interface or RS-232C interface. The price is $150.

Macrotronics, Inc., 1125 N. Golden State Blvd., Turlock, CA 95380. Reader service number 496.

Financial Analysis and Modeling Package

Desktop/Plan II is a professional financial planning software package for Apple II personal computers, with Applesoft, that speeds the preparation of financial statements, budgets, forecasts, projections and analyses. Desktop/Plan II offers such new features as high-resolution graphics plotting, the capability of receiving data from Personal Software's VisiCalc program and a moving cursor which helps simplify data entry. Because the Desktop/Plan II stores its formulas and values on diskette, a financial model larger than the computer's memory can be constructed. Package, including backup disk, $199.95.

Personal Software, Inc., 1330 Bordeaux Drive, Sunnyvale, CA 94086. Reader service number 463.

PL/1 Language for Desk Top Computers

Digital Research's new PL/1-80 is available from Westico, Inc., 25 Van Zant St., Norwalk, CT 06855. PL/1-80 is an all-purpose application programming language for 8080, Z-80 and 8085 microprocessors. PL/1-80 programming systems comprise four major components: PL/1-80 compiler, Link-80 linkage editor, PL/1-80 run time library and RMAC relocatable macro assembler. The system generates industry standard Microsoft relocatable code, so users can link load subroutines created by other language translators.

The package includes sample programs, three manuals and a programmer's quick reference guide. Price is $500. Reader service number 497.

Cheaptalk for the TRS-80

Cheaptalk enables the TRS-80 to talk; a small audio amplifier connected to the cassette output plug is all that's needed. Instructions are supplied for modifying your CTR-41 for use as an audio amplifier. Spoken words are digitized and stored in memory as complete self-contained subroutines callable by the BASIC SR statement or a Z-80 assembly-language call. Each word requires 512 bytes of storage, and you can store as many words as you have memory for. Don't expect stereo-quality audio, but it is intelligible. Cheaptalk cassette, $19.95.

Alan Saville, PO Box 5190, San Diego, CA 92105. Reader service number 499.

Payroll Program for TI 99/4

Charles Mann & Associates, Micro Software Division, 7594 San Remo Trail, Yucca Valley, CA 92284, has announced Payroll I for the TI 99/4 computer. The system allows small businesses to operate a full weekly, semi-monthly or monthly payroll system. It calculates and records all federal taxes and FICA, and three user-formulated fields for state taxes, insurance, union dues and so on. The system can handle salaried, commissioned and hourly employees. It provides total retention of all individual payroll entries and a built-in quarterly audit balancing routine.

Payroll requires a single disk drive, a small micro printer, Extended BASIC and a 16K CPU. The program costs $199.95. Reader service number 487.

Demo Package

Small Business Systems Group, Inc., 6 Carlisle Road, Westford, MA 01886, is offering a demo package of its TRS-80 Model II software. Samples of A/R, A/P, G/L, Invoicing, Inventory, Payroll, Client Billing and Name and Address-III are included. Variable speed capability allows a salesperson time to verbally address each screen. Dealers will find the unattended mode useful as an eye catcher. The package includes the SBSG catalog, demo diskette and manual. The price is $50. Reader service number 493.

Word Processing Tools

The Word is a complete toolkit of programs for spelling checking and correction, word counting and other word processing functions. Compatible with CP/M 1.4 and 2.2, The Word works in conjunction with any CP/M text editor or word processor. It contains a 40,000 word dictionary, and will find and list spelling errors in any text file. Other included programs will flag the misspelled words for easy correction with your text editor, locate possible correct spellings automatically and help you make and solve crossword puzzles. Eight-inch disk and manual are $75.

Oasis Systems, 2765 Reynard Way, San Diego, CA 92103. Reader service number 489.

Apple Sacks

New software packages for the Apple II are available from ComputerMat, Box 1664F, Lake Havasu City, AZ 86403. Apple Sack I contains 40 Applesoft programs, including a selection of business, game, utility and home applications. Apple Sack II is a collection of TV trivia questions, and is written in Integer BASIC. Apple Sack III contains eight adventures and simulations from the old west to the new space frontier, available in Integer and Applesoft. Apple Sack IV contains utility programs allowing you to fix, catalog, copy, sort, transfer, find, remove and perform other functions with your Apple, and is available in binary, and Integer and Applesoft BASICs. Each sack is $24.95. Reader service number 495.

New TRS-80 Operating System

LDOS is an operating system that bridges the gap between TRS-80 microcomputers: compatibility with Model I is available now; compatibility with Models II and III will be ready soon. LDOS is fully device-independent, supporting five- or eight-inch floppy disks, double- or single-sided, double- or single-density and hard disks. It provides keyboard type-ahead, auto-repeat and a keystroke multiply feature that allows users to define phrases for each alphabetic key. It contains all standard library commands and an extended library, extended utilities, enhancements to BASIC and its own set of device drivers.

Lobo Drives International, 354 South Fairview Ave., Goleta, CA 93117. Reader service number 494.

Smart Terminal/File Transfer Program

Crosstalk is a smart terminal and file transfer program for the Hayes Microcomputer Products Micromodem 100, and is available for CP/M or North Star DOS. The program acts like a sophisticated smart terminal, allowing the user to capture data from a remote computer system and save it to a disk file, or to send a disk file to the modem. The program features auto-dial and redial; auto-answer; selectable baud rate, word length, parity and duplex; concurrent printer and video; user-definable break key; and disk directory command. It uses command files to store setup information for frequently called numbers. Price is $75.

The Microstuf Company, Box 33337, Decatur, GA 30033. Reader service number 498.

Genealogy Program for Heath Computers

Commsoft, 665 Maybell Ave., Palo Alto, CA 94306, is introducing Roots 89: A Software Tool to Track Your Ancestors. Roots 89 can store, retrieve and display thousands of facts about your family. It is seven linked programs designed to list family members, display and print family group sheets and pedigree charts, compute relationships and show important anniversaries for each day of the year. The documentation provides an extensive course in genealogy. Price is $124.95. Reader service number 464.

Word Processor for TI Systems

Word-990 is an interactive word processor that lets you see your document exactly as it will appear when printed. It features word fill, justify, insert, delete, overtype, center, extract, search, append and many other commands. Word-990 runs in SBC BASIC on the Texas Instruments DS990 computer series, and is priced at $495.

Synergistic Systems, Cobble Hill Road, E. Thetford, VT 05043. Reader service number 491.

Upgrade Your DiskAte Operating System

DocuMint was developed by Responsive Logic, 156 Donald St., Oregon City, OR 97045, to improve the appearance of text-edited copy. It works with the Thinker Toy Discus 1 and DiskAte operating system. DocuMint produces true proportional spacing by changing the size of spaces between characters, and is used with a proportional space printer such as the Centronics 737. It permits automatic pagination, standard or customized formatting and variable line spacing. DocuMint is entirely menu driven. The eight-inch single-density disk costs $85. Reader service number 490.

North Star Enhancement

N*Sort is an addition to North Star BASIC that allows programmers to sort one- and two-dimensional arrays or strings with a simple BASIC statement, such as "SRT A$, LEN(A$),1". N*Sort incorporates one-level sort keys for both numeric and string sorts, and is implemented as a Shell-Metzner machine-language routine that is about 1800 percent faster than an equivalent algorithm in BASIC. The tandem version coloads with BASIC, and the ROMable version places N*Sort outside BASIC's memory region. The package includes diskette, manual and BASIC program which interfaces N*Sort to any release 4–5.2 North Star BASIC. Price is $89, plus shipping.

SZ Software Systems, 1269 Rubio Vista Road, Altadena, CA 91001. Reader service number 492.

Software Tools for CP/M

Programming languages and other software tools for CP/M are now available, on standard eight-inch single-density disks, from The Software Toolworks, 14478 Glorietta Drive, Sherman Oaks, CA 91423. The C/80 C Compiler implements a large subset of the C programming language. LISP/80 is an 8080 implementation of the well-known artificial intelligence language. Structured programming constructs for Microsoft Fortran are provided by the RATFOR preprocessor. It comes with complete source code, written

in RATFOR, and sample programs. The Text program is a versatile text formatter. When used with a good file editor, Text supplies word processing capability usually found only in much costlier systems. Prices range from $20–$40. Reader service number 465.

Foreign Language Word Processor

Hebrew II is a useful tool for education and everyday communication. You can type and edit a page at a time of Hebrew text. You can also use it to label graphs, maps and pictures on the Apple II screen and then print them. The finished text or picture may be saved to disk and/or printed on the Silentype printer. In accordance with common usage, vowels are not provided at this time. Hebrew II is perfect for teaching students Hebrew spelling and typing. Printed pages can be submitted to the teacher and the page can later be edited on the screen to show the recommended corrections. It requires 48K of RAM, Applesoft in ROM and one disk drive. Price is $60.

Michael Heckman, 11 S. Hancock St., Madison, WI 53703. Reader service number 462.

Asset Management System

UNIDEP is an asset management system. It calculates depreciation, investment credit and bonus depreciation. It performs automatic subtotaling, as selected by the user. It also files information about each asset. With UNIDEP you can do projections to experiment with various options. The user chooses the relevant options, including depreciation rate, term, length of reporting period, when and how many periods to print and when to update information. It runs on a 32K Apple or 64K TRS-80 Model II, with disk drive. Price is $99.

Natural Language Systems, 411 Barber Ave., Ann Arbor, MI 48103. Reader service number 486.

MBC SYSTEMS inc.

(203) 342-2747 TWX 710 4286345

COMPUTERS

HEWLETT-PACKARD
- HP-85A $2795
- HP-83A $1990

NORTH STAR
- HRZ-2-32K-DD-FACTORY ASSEMBLED $2795
- HRZ-2-32K-QD-FACTORY ASSEMBLED $2995

COMMODORE BUSINESS MACHINES
- 8032 (80) CHARACTER SCREEN .... $CALL
- 4032 (40) CHARACTER SCREEN .... $CALL
- 8050 DUAL DRIVE 1MEG STORAGE .... $CALL
- 4040 DUAL DRIVE .... $CALL
- CALL US FOR COMPLETE LINE

INTERTEC SUPERBRAIN
- SUPERBRAIN 64K 2D $2890
- SUPERBRAIN 64K 2Q $3490

ALTOS, APPLE, ATARI ETC. ALSO IN STOCK, CALL FOR PRICES!

PRINTERS

- DIABLO 630 .... $2395
- NEC 5510 SERIAL OR 5530 PARALLEL $2595
- C.ITOH $1775
- ANADEX 9500 OR 9501 .... $1490
- IDS PAPER TIGER 445G $820
- IDS PAPER TIGER 460G $1250
- IDS PAPER TIGER 560G 132 COL '5" PAPER $1395
- WE CARRY CENTRONICS COMPLETE LINE, ... $CALL
- EPSON MX-80FT FRICTION & TRACTOR $695
- EPSON MX-80 .... $595
- EPSON MX-70 .... $425
- OKIDATA MICROLINE 80 .... $499
- OKIDATA MICROLINE 82 .... $690

TERMINALS

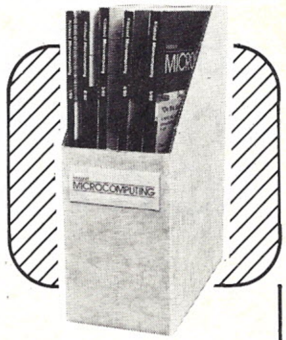
- TELEVIDEO MODEL 950 .... $1075
- TELEVIDEO MODEL 920C $890
- HAZELTINE MODEL 1400 .... $599
- HAZELTINE MODEL 1420 .... $890
- INTERTUBE AND EMULATOR $775
- SOROC IQ 120 .... $850
- SOROC IQ 135 .... $899

SINCE 1977 COMPLETE SALES AND SERVICE. VISA & MC OK!
PRICES ARE SUBJECT TO CHANGE, MOST ITEMS IN STOCK.

Multi-Business Computer Systems ✓81

28 MARLBOROUGH ST. PORTLAND, CT 06480

is HARD COPY STORAGE a problem?



KILOBAUD MICROCOMPUTING, as thick as it is, is more like a floppy when it comes to standing on the bookshelf. Try the **KILOBAUD MICROCOMPUTING** Library Shelf Boxes . . . sturdy corrugated white dirt-resistant cardboard boxes which will keep them from flopping around. We have self-sticking labels for the boxes, too, not only for **KILOBAUD MICROCOMPUTING,** but also for **73 Magazine, 80 MICROCOMPUTING** . . . and for **CQ, QST, Ham Radio, Personal Computing, Radio Electronics, Interface Age,** and **Byte**. Ask for whatever stickers you want with your box order. They hold a full year of **KILOBAUD MICROCOMPUTING, 80 MICROCOMPUTING** . . . or **73 Magazine**. Your magazine library is your prime reference; keep it handy and keep it neat with these strong library shelf boxes. One box (BX-1000) is $2.00, 2-7 boxes (BX-1001) are $1.50 each, and eight or more boxes (BX-1002) are $1.25 each. Be sure to specify which labels we should send. Have your credit card handy and call our toll-free order number 800-258-5473, or use the order card in the back of the magazine and mail to:

kilobaud MICROCOMPUTING™
peterborough nh 03458

GET ORGANIZED

With These Utility Programs From Instant Software

FIND IT QUICK

Put an end to the Misplaced Information Syndrome! Here is a reliable, fast, subject-oriented information locator—FIND IT QUICK! This information indexing and retrieval system is versatile and flexible—the ideal tool for doctors, lawyers, engineers, businessmen, educators, anyone who needs to locate information.

Designed with a journal/magazine format, FIQ is versatile enough to be used to store and retrieve any type of reference information including client lists, vendor advertisements, report bibliographies and more—even poems and famous quotations. Subjects can be indexed or called according to author, journal and date or by up to nine keywords that you determine. Keywords can be used singly or in combination during the search routines.

FIQ can be used by anyone. It requires no special skill, just the ability to type. Article references to be stored can be preselected and then entered into the system by a secretary or assistant. The contents of entire file cabinets can even be indexed and stored!

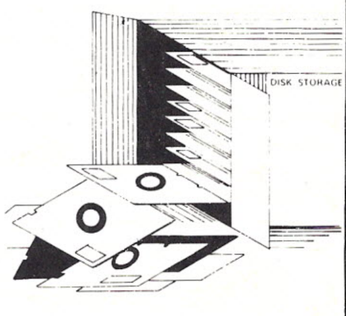
FIND IT QUICK requires a minimum system consisting of a TRS-80 Model I Level II with 16K of RAM, an Expansion Interface with 16K of RAM and at least one disk drive. Up to 1080 items can be stored on one data disk. Instant Software's Tiny DOS operating system is included on the program disk so that you can use the program without fuss or bother.

Make your computer work like your own personal librarian with FIND IT QUICK. Can you afford to wait another day?

Order No. 0258RD $49.95. Disk

MASTER DIRECTORY

The MASTER DIRECTORY is a disk file storage program that reads the files on all your disks and stores the file names and extensions and even records the free space on each disk. All you have to do is number the disks in your library and the MASTER DIRECTORY will keep track of their contents. You can read the names, displayed alphabetically, search the DIRECTORY for file names and extensions, delete disks and search for free space. You can store 5000 files or 320 disks, whichever comes first.



Your disk storage problems are over now that the MASTER DIRECTORY is here. This package requires the following minimum system:

1. A TRS-80 Level II with 16K RAM.
2. An Expansion Interface.
3. One (or more) disk drives.

Order No. 5005RD (disk-based version) $29.95

✓40

Instant Software™
PETERBOROUGH, N.H. 03458

TO ORDER:
SEE YOUR LOCAL
INSTANT SOFTWARE DEALER
OR CALL TOLL-FREE
1-800-258-5473

6800 Tape-Based Sort Processor
UniFLEX Multi-User System
ACIA Prototyping Board

Dr. Preble's Programs' LDP-3

While many system designers feel that only a dual-disk, 48K system is adequate for the small-business user, there is much that can be done with a small tape-based system. It may not allow a full-feature general ledger, billing package or high-speed program loading, but the cassette system will perform many tasks for the home and small-business owner at a reasonable cost.

A small system can easily be used as a mailing list manager. Dr. Preble's Programs (102 W. Madison St., La Grange, KY 40031) is offering a fast sort program for the 6800. The LDP-3 is priced at $14.95 and is supplied in both source and object on an AC-30-compatible tape. The user's manual contains full operating instructions and a commented source listing.

The LDP-3 is a machine language sorting program that can easily be adapted to almost any 6800 system. The program itself takes less than 400 (hex) bytes of memory. While a 16K system will allow 150 addresses to be sorted, even an 8K system will perform a useful task. The more user memory available, the more items the LDP-3 will sort.

The Sort Processor is designed to sort files created with a text editor. The TSC Text Editor is the ideal companion to the LDP-3. It is easily obtainable and is reliable.

The editor creates a file of addresses and names up to 256 characters per line. A typical address line is:

GRAPHICS I,345 MAIN ST,RACINE WI,.53403

At the beginning of the file, before the first name, a special start-of-file character occupies one line. This can be any non-alphanumeric character such as the up arrow (↑). This signals the LDP-3 that the file is starting. The last line created by the editor is an end-of-file character. Again, this can be any unused character. It tells the LDP-3 that the end of the file has been reached. The editor is used to save the address file on a cassette tape.

The Sort Processor is loaded and the tape file is input and sorted. The sorted file can be listed on the terminal and saved to tape, where it can be used to print a sorted address list using BASIC, an editor or a short assembly-language printer-driver routine.

The file can be sorted by fields; that is, by name, address, city/state or zip code. All that is required is to insert a sort header before that desired field. In the above example, the period (.) is used to identify the zip code field. The character can be placed before any field in the line. Since the Postal Service requires bulk mailings to be sorted by zip codes, the LDP-3 can create an address file sorted by zips. These will be in order when they are printed as address labels. Even with only 150 pieces of mail to sort, the LDP-3 will save time.

If your lists include both upper and lowercase items, the LDP-3 will convert the lowercase to uppercase for the sort. This will prevent an item such as Able from being sorted before aardvark. In ASCII, uppercase letters have a lower value than lowercase letters.

With a 300 baud system, it takes about 60 seconds to load 50 names and addresses from a cassette. These can be sorted by name or zip code in about 12 seconds. While the LDP-3 does not have the program loading time of a disk system, its cost is within the reach of almost everyone.

Since the LDP-3 can sort almost any type of file, it has many uses and can be used for more than address sorting. If your computer club's roster is on a tape file, it can be sorted alphabetically by name, address, system or special interests. It would be easy to maintain a file of your library or record collection and keep it sorted by title, author or subject. The more you use the LDP-3 Sort Processor, the more uses you will find for it.

As well as sorting a file from tape, the LDP-3 will also sort any continuous block of memory. It should not be difficult to reassemble the Sort Processor in any convenient memory location so it is co-resident with an editor or BASIC.

The program is self-documenting while running. It asks the user for the parameters and proceeds with the sort. It is easy to sort a file first by name and then by zip codes.

When the LDP-3 is started, the user is asked to supply the following parameters he selected: the record separator (usually a carriage return), the sort header (for a field sort), the start-of-file and end-of-file characters, perform the search from tape input or memory and uppercase conversion. If a long file is to be sorted, a verification option is available. This prints a plus sign (+) on the terminal for each cycle of the sort.

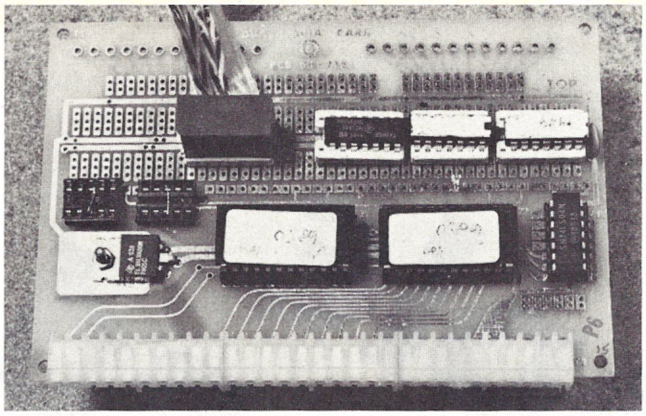
Do you need the LDP-3? If you do not use a disk and have lists that need to be alphabetized and saved, by all means, the LDP-3 Sort Processor will be a useful addition to your system.

Dennis Doonan
Racine, WI

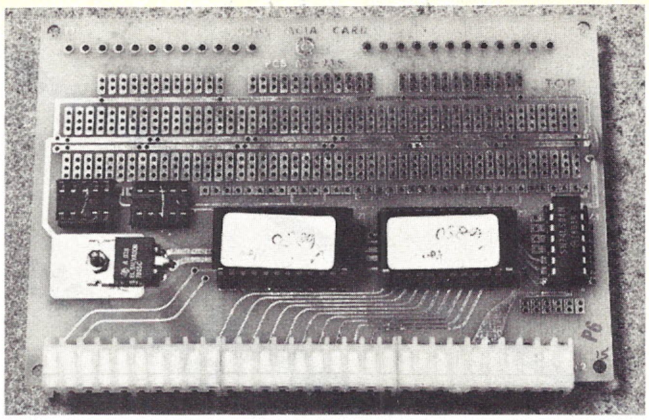
UniFLEX

Today's technology makes it possible to develop software comparable to that for much larger installations. The UniFLEX operating system by Technical Systems Consultants (PO Box 2570, West Lafayette, IN 47906) takes advantage of the power of Motorola's 6809, and will also be available for the 68000.

UniFLEX is based primarily on UNIX, an operating system developed by Bell



The QRC 80-215 in use as a dual RS-232 serial interface.



The QRC 80-215 assembled and ready to use for prototyping.

Laboratories. UniFLEX was not designed for the hobbyist or very small business with only one or two users; it is a true multitasking, multi-user system, allowing up to 12 users under the 6809 version. It offers an environment similar to that of larger systems, where each user is assigned a name and password required to log into the system. A complete log of user activity makes UniFLEX ideal for an educational institution, where such activity should be monitored.

Being a true multi-user system means that each user can be running a totally different program. For example, one user may be doing word processing, while another user is assembling a source file. This ability makes UniFLEX well suited to the office environment.

Technical Systems Consultants will write custom versions of UniFLEX for nonstandard systems, so the hardware configuration may vary. The standard version requires at least one video display terminal with lowercase capability, one serial interface card, a Southwest Technical Products computer system with an S/09 mainframe with 128K of memory and the DMAF-2 eight-inch disk system. More RAM, serial interfaces and terminals are supported, as well as the CDS-1 hard disk system by SWTP.

Technical Systems Consultants offers their major software in versions compatible with UniFLEX. Their editor and assembler are included with the initial purchase of UniFLEX, and a multi-user version of their high-speed BASIC interpreter is also available. Anyone who buys UniFLEX should not fear the lack of software support; other software companies are already offering compatible software.

In addition to multi-user operation, UniFLEX has two other important features. First, it can run background jobs. An operator can, for example, print a company's payroll checks using a program written in the BASIC interpreter and be editing text at the same time, by running BASIC under a background task.

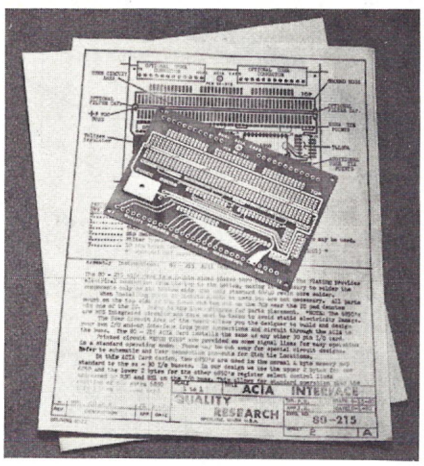
This can lead to a problem which is solved by this second feature: UniFLEX can swap to an external device. Swapping can be done with the eight-inch disk system, with some loss in speed. With the hard disk, the operation is almost invisible to the user. Swapping, in effect, gives the system more memory to do its multi-tasking operations. Of course, if speed is really a problem, more memory can be added to the system that will be used for these swapping operations.

In the future, more companies will be offering operating systems based on UNIX by Bell Laboratories. Those with eight-bit microcomputers will probably be out of luck, because most of them lack the sophistication required to develop such a system. Technical Systems Consultants even admitted that no attempt at a 6800 version of UniFLEX will be made. But in the 16-bit market, this kind of software can be expected to flourish, and UniFLEX will certainly be the model for all those who follow.

Tim Chase
Milwaukee, WI

QRC ACIA Board

Quality Research Co. (Box 7207, Spokane, WA 99207) offers a convenient way to design a custom, dual-serial interface for your SS-50 system. The 80-215 ACIA card is a prototyping board designed to fit one 30-pin I/O slot. It is a quality, double-sided, plated-through board with data and control lines etched for two 6850 ACIAs.



The QRC 80-215 ACIA prototyping board and documentation as received.

The following signals are brought to an ample prototyping area: I/O select, R/W, Ø2 clock, reset, IRQ, NMI, user defined and index lines, ±12 V, ground and the onboard regulated +5 V lines. Baud clocks, TX and RX clocks, and the TX and RX data lines are separate for each ACIA. Baud rates and interrupts can be selected for each 6850 by jumpers. There are five inverter gates available for user design. It will be easy to design a dual RS-232, current loop, or TTL interface or any combination on a single board. The 80-215 is set in the standard operating configuration and is easily changed by the user.

Two 6850 ACIAs occupy the four-byte set of address locations used on the SS-50's I/O bus. This allows conventional operation with a second ACIA on the board. The upper two address bytes are used for one 6850. The lower two bytes are used for the second 6850's register select lines controlled by RS0 and RS1 on the I/O bus. With an SWTP 6800 and SWTBUG, the QRC board cannot be used on the control port. SWTBUG expects both address pairs of port 1 for a single ACIA. Other than that, the 80-215 can be used on any port to allow two terminals, printers, modems or any combination to occupy a single I/O slot.

Although there is no solder masking or component labels, assembly is easy and only takes a few minutes. The documentation includes assembly information, a component layout and a schematic.

Shipment was prompt and service personal. The 80-215 bare board is $8.50 and the bottom edge connectors are $3 postpaid. If your design requires a 6820 parallel interface, QRC offers the 80-205 PIA prototyping card (see *Microcomputing*, July 1980, p. 14, for details).

Dennis Doonan
Racine, WI

✓See List of Advertisers on page 210

Computerize Your Small Business
A/D and D/A Interfacing
Computers for Home Use
Z8000 Assembly Language Programming

How to Computerize Your Small Business

Jules A. Cohen
Prentice-Hall, Inc., 1980
Paperback, 171 pp., $7.95

Even though this book is not concerned with *micro*computers for the *very* small business, it should be required reading for anyone trying to sell computers to businessmen. It gives necessary insight into the thought processes and planning of the business owner and, if he is smart enough to hire one, his consultant.

The author begins with an explanation of data processing and the computer components required, without mentioning specific makes and models. This ensures that the book is not obsolete by the time you get it. He then goes through the steps involved in analyzing the system requirements, writing a request for proposals, avoiding being sidetracked by fast-talking salesmen and creating an evaluation matrix to be used for the buy decision.

Two specific system examples are given. The first is for a business accounting application, the second for a large word processing system. As Cohen points out, the large system might be better implemented by a collection of smaller systems (enter the microcomputer).

About a third of this small manual is taken up with the description of the system requirements for an eight-user word processor with database management functions. Included are a lot of useful boilerplate, report formats, file organizations and CRT screen layouts for the input operators. These forms alone are worth the modest price of the book. They can provide a computer salesman with graphic examples of how a system should be organized and what it can do for the customer.

Whether you are thinking about or planning a computer system for your own business, advising a businessman on computer selection or trying to sell a business computer, you should read this book. When you are done, keep it handy to show the potential customer. You might have to explain that microcomputers differ from the examples given, but that the difference is only one of scale. The planning, analysis and evaluation decisions that are the subjects of the book apply equally well to systems large or small.

Overall, this is a most informative book, logically organized, and both easy and interesting to read. What we need now is for someone to write a similar book aimed specifically at microcomputers for the tiny business, accurately explaining their capabilities and limitations. Properly done, this could be a big seller.

Until then, I'll keep this volume handy as the best "how-to-computerize" text so far.

Ken Barbier
Borrego Springs, CA

Microcomputer Interfacing Handbook: A/D and D/A

Joseph J. Carr
Tab Books, Inc.
Blue Ridge Summit, PA
Softbound, 350 pp., $8.95

"If you've been itching to do more with your microcomputer than just play games, this thorough introduction to using analog-to-digital and digital-to-analog converters has all the answers you want. This big new manual not only tells you how to interface both A/D and D/A converters with all kinds of microcomputers—it shows you how to do it as well."

The above blurb from the back cover of this book is as good a two-sentence summary of the book as can be given. This book is *complete*. It treats virtually every aspect of digital and analog interfacing. Since we live in an analog world, such interfacing is necessary to measure or control analog quantities with our digital computers.

I would not recommend this book as a first text in analog conversion. Much of the material assumes a familiarity with electronics, especially op amps and TTL circuitry. I personally found the material easy, but then, I teach electronics to college seniors and graduate students. A knowledge of algebra is a must, and a very small amount of calculus is used.

There are chapters which are tutorials on subjects in electronics, such as an entire chapter on the basics of op amps and a chapter on analog switches. Both of these speak in simple terms, pointing out the salient features of each, and leading the student through the material on a step-by-step basis, with example calculations.

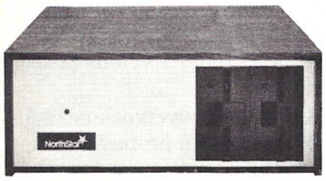
These chapters are painfully short, though. For example, there is not much discussion on the limitations of op amps in ADCs and DACs, such as slew rate and frequency response (I learned the hard way that an LM741 isn't fast enough to substitute for an LM310), and the discussion of op amps as comparators could be beefed up, since comparators are the heart of many ADCs.

The writing style is mixed. Portions of the text are very easy, giving step-by-step instructions through the basics, and then become quite technical in the heavier topics. The book is really more oriented toward A/D and D/A design than as a beginning textbook for the person who wants to measure or control analog voltages with a digital source. A beginner would find this book tough going—it's not for the person who just wants a circuit he can whack together without understanding what it does.

For someone who is familiar with electronics using integrated circuits, even on a black-box level, the book is quite useful.

LOWEST PRICE - BEST QUALITY

NORTH STAR



North Star Horizon 2

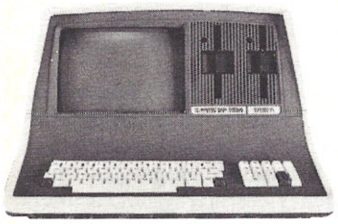
2-5¼ Disk Drives
32K Double Den
Factory assem. & tested
Factory guaranteed
List 3095

only **$2274**

POWERFUL NORTH STAR BASIC FREE
SUPERB FOR BUSINESS & SCIENCE

FACTORY ASSEMBLED & TESTED	LIST	ONLY
HORIZON-1-32K-DOUBLE DEN	$2695	$1980
HORIZON-2-32K-QUAD DENSITY	3595	2674
HORIZON-2-64K-QUAD + HARD DISK	9329	7149
HORIZON RAM ASSM	16K = $389.	32K = $579
HORIZON RAM KIT	SALE! 16K = $314	32K = $469
HORIZON DISK DRIVE SALE DOUB DEN	SAVE!	315
NORTH STAR HARD DISK 18 Mb	4999	$3929
PASCAL-PLUS 14,18 OR 36 DIGIT PRECISION		249

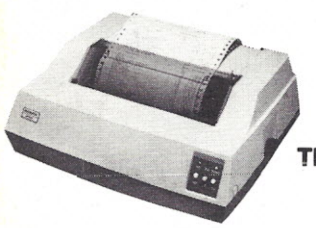
SUPERBRAIN
ZENITH



SUPERBRAIN QD 64K
List $3995 only $2995

Z-89 48K
List $2895 only $2299

TERMINALS Z-19 $725
INTERTUBE III only **$725**
DIP-81 PRINTER only **$395**



MICROTEK $675

NEC PRINTER $2569
TRACTOR, THIMBLE, RIBBON

InterSystems

ITHACA INTERSYSTEMS 2A

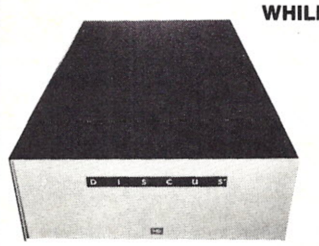
Z-80A CPU 4 MHz
64K Dynamic RAM
Front panel
V I/O—with interrupts
FDCII Disk Controller
20 slot motherboard

List $3595 You Pay Only **$2795**

PASCAL/Z + THE FASTEST PASCAL $375
GET READY FOR ITHACA'S Z-8000
8086 16 BIT CPU & SUPPORT CARD SEATTLE $575

MORROW 8" DISK

DISCUS 2D + CP/M® 600K ONLY $938
DISCUS 2 + 2 + CP/M® 1.2 MEGA B. $1259
ADD DRIVES 2D = $650 2 + 2 = $975
2D-DUAL + CP/M® + MICROSOFT BASIC = $1555
$2293 VALUE OVER 32% OFF!
WHILE THEY LAST



MORROW HARD DISK
26,000,000 BYTES!!
LIST $4995 ONLY $3995
CP/M® IS INCLUDED!

SAVE ON MEMORY AND PROGRAMS

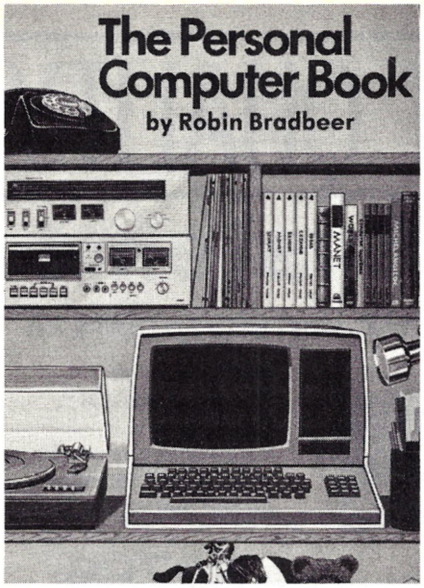
SYSTEMS MEMORY 64K A & T 4mHz	$599
SYSTEMS MEMORY 64K BANK SELECT	$789
CENTRAL DATA 64K RAM	$599
ITHACA MEMORY 8/16-bit 64K	$845
SEATTLE MEMORY 8/16 BIT 16K 4Mhz	$275
SSM KITS Z-80 CPU	$221
VIDEO BRD V83 4Mhz	$412
ANADEX PRINTER DP-9500-1	$1349
CAT NOVATION MODEM	$169
TARBELL DISK CONTROLLER DD	$445
ECONORAM 2A 8K ASSM	$179
NSSE 1-22 & P01 TERRIFIC PROGRAMS	ONLY $10. EACH
NORTHWORD 294 MAILMAN 234	
INFOMAN	$364
TARBELL COMPUTER-PHONE	
RCA-COSMAC VP-111 99 RCA-COSMAC VP-711	$199
COLOR! RAINBOW 385	
SPECTRUM	$289
EZ-80 Machine Language Tutor	$25
EZ-CODER Translates English to BASIC	$71
ECOSOFT FULL ACCOUNTING PKG	$315
BOX OF DISKETTES	$29
SECRETARY WORD PROCESSOR	The Best! $99
TEXTWRITER III Book Writing Program	$112
GOFAST NORTH STAR BASIC Speeder	Upper $71
Which Computers are BEST? BROCHURE	FREE
North Star Documentation refundable	w/HRZ $20

AMERICAN SQUARE COMPUTERS BEATS ADV. PRICES

American square Computers ✓56

919-889-4577 KIVETT DR. JAMESTOWN N.C. 27282 919-883-1105

® CP/M is a registered trademark of Digital Research, Inc.



Indeed, it could be an invaluable reference! The major selling point of this text is its completeness. Everywhere you look there are diagrams, examples of use and sample calculations.

One point I like about the book is that it begins with a discussion of analog and digital signals. Oftentimes the conversion method used depends on the signals available or desired.

There is a complete comparison of A/D and D/A conversion methods over several chapters, giving the advantages and disadvantages of each, how they work, and how well they work. Included with this is a discussion of precision, reference voltages, sample and hold errors, multiplexing, sources of error, limitations to precision and the like.

There is an entire section on analog to digital conversion rates which is very good. It tends to be application-oriented, as well. The thrust of A/D conversion is toward conversion of dc voltages, or of ac sampling and digitizing. Specific examples are given in the cases of dc voltmeters or digitizing of an electrocardiogram and Fourier synthesizing. A point I find lacking, mostly because I do a lot of it in my work, is fast pulse-height analysis (PHA), where the amplitude of a pulse only 2 us wide must be digitized several thousand times a second, at random times. The material is there, but requires some searching.

The rest of the book is devoted to specific examples and applications. There is an analysis of many major D/A and A/D chips, with application circuits based on these. The book provides pin-out and specifications. This is helpful in that the reader won't have to reinvent the wheel, but limiting in that the application presented may not be the reader's application. There are no examples of a converter built up from scratch, other than in general terms.

D/A applications covered are motor control, wave-form generators and an X-Y display.

Two chapters cover interfacing with microprocessors. The D/A chapter is mostly a discussion of address decoding, while the A/D chapter goes into Tri-state logic, data buses and a long explanation of the use of interrupts for on-demand processing. There is also a chapter on software data conversion techniques which will be of great interest to the hardware-poor computer hobbyist who nonetheless likes to "program around" a problem and who would like to try his hand at analog measurement control.

When I evaluate a book for adoption to one of my college classes, I ask myself three questions:

1) Is it complete—does it cover the material I want?

2) Does it have lots of examples and diagrams? and

3) Will it be understandable to my students?

In the case of this book, the answer to questions 1 and 2 is an emphatic "yes." Question 3 gets a qualified "maybe." It's not for someone who doesn't know an op amp from a NAND gate, or for the games programmer who's never had the cover off his TRS-80. But for the experts who read *Microcomputing*, or for the intermediate-level hardware hacker, the book contains a wealth of information, and is highly recommended.

Dr. Gordon Wolfe
Oxford, MS

The Personal Computer Book

Robin Bradbeer
Input Two-Nine
Bradford, England, 1980
Paperback, 220 pp.

Owning Your Own Computer

Robert L. Perry
Everest House
New York, NY, 1980
$10.95 paperback, $15 hardcover; 224 pp.

The back cover of *The Personal Computer Book* says that Bradbeer is a "free-lance writer on personal computing." Is this anything like being a free-lance writer on drugs? Apparently so, if this book is any indication. Bradbeer talks about computers with as much restraint as Timothy Leary discussing sunshine acid.

According to Bradbeer, micros will soon be "as natural a part of the home as the TV set is today." Not one to shy away from the challenges of redundancy, he says five pages later that we will have a "total domestic information and data processing system . . . which we will come to find just as natural as today's electric cooker, washing machine and colour TV."

These are absurd and premature assumptions. In the first place, electric cookers, washing machines and color TVs are not a natural part of millions of homes, and to state so is to perpetuate a middle-class fallacy. In the second place, the microcomputer is nowhere near being the plug-in appliance that Bradbeer's remarks suggest. In the third place, the consumer has yet to be convinced that microcomputers are the necessity that many micro enthusiasts would like to believe.

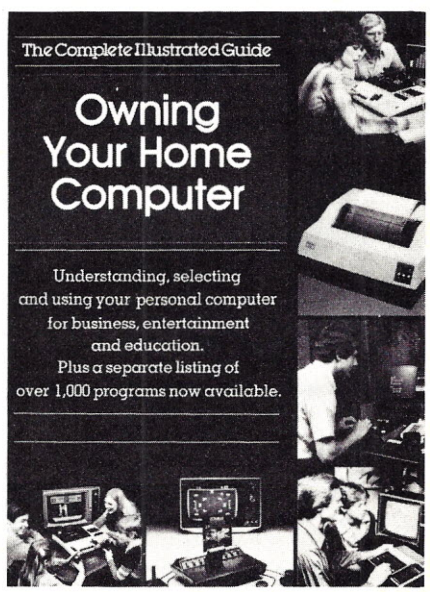
Bradbeer's attitude reflects one glaring fact—micros are not yet able to sell themselves. They need people to keep reminding the public of all the marvelous tasks they can perform.

Bradbeer's irritating overenthusiasm is the least of this book's problems. Unfortunately, the book is also a prime example of the lamentable decline in quality book writing and editing.

Bradbeer's grammar is atrocious. Witness this example: "The last thing wanted is for your supplier, who you will depend on a great deal to start off with, not to go bust just when you need him!" This sentence is so convoluted that it means exactly the opposite of what it's supposed to. The entire book is written in this awkward style. It certainly proves that Americans aren't the only ones who can massacre the language.

What's more, the book is riddled with inconsistencies and typographical errors. Bradbeer freely alternates, for example, between "Basic" and "BASIC," sometimes on the same page. A section on interfaces refers twice to the "IEEE-4888" before the mysterious extra 8 disappears.

As for the content, you'll find no surprises here. Bradbeer covers all the usual ground—how a computer works, binary numbers, computer languages, input and output devices, storage and printers. He also provides a list of manufacturers, distributors and computer clubs in the U.K. and a list of English-written magazines. And, of course, he includes a chap-



ter on the various micros and peripherals on the market, which was outdated before the book even hit the press.

But ultimately, it's Bradbeer's ungodly English that sinks this book. Perhaps someone at Input Two-Nine will have the decency to edit it before the second edition comes out.

Robert L. Perry's *Owning Your Own Computer* suffers from the same kind of hyperbole that Bradbeer's book does. Perry, too, wants the reader to know about "all the fantastic ways to use a home computer," and predicts that in 1995, "we'll wonder how we ever lived the way we do without our home computer." But his book is more extensive, and his writing is the work of a professional.

Perry focuses much of his attention on the broader and more significant micro applications—namely, those related to information networking. In fact, it takes him only three pages before he's discussing teletext, viewdata, The Source, CompuServe and the Electronic Information Exchange System. He explains these uses in detail in a later chapter. Perry seems to recognize that information retrieval is what will ultimately sell the home computer.

Like Bradbeer, Perry includes the standard material on how computers work. He also devotes a chapter to a short history of computers, another to buying a micro, and two more to micros in education and at work. One short chapter covers the uses of micros in helping the handicapped.

Unfortunately, Perry waters his book down with a 67-page rundown of the major microcomputers. Again, while these descriptions might give the reader an idea of what kinds of micros are on the market, the section is already out-of-date and incomplete. He also includes quite a lot of filler material: a silly chapter entitled "Ninety-nine Common Things to Do with a Home Computer," an obligatory gaze into the crystal ball and an appendix listing 1050 computer programs.

Despite its flaws, *Owning Your Own Computer* is an adequate introduction for the newcomer. If anything, it provides too much information, but the discriminating reader should be able to pick out what he needs.

The Personal Computer Book, on the other hand, will be useful only to those with plenty of patience and deciphering skills.

Eric Maloney
***Microcomputing* Staff**

Z8000 Assembly Language Programming

Lance A. Leventhal, Adam Osborne and Chuck Collins
Osborne/McGraw-Hill, 1980
Softcover, 898 pp.

Osborne/McGraw Hill is probably one of the most experienced documentation companies around. Adam Osborne's original organization was among the first to produce microprocessor/microcomputer books.

And it shows. *Z8000 Assembly Language Programming* is another in a long line of highly descriptive books. Using Osborne's typical typesetting feature (important information is given in **boldface** type), this book explores the ins and outs of Z8000 programming.

The book is divided into 17 chapters. The first chapter contains an introduction to assembly-language programming, and a brief discussion of high-level languages. The second chapter describes the features of an assembler. Both of these chapters could have been left out—they seem somewhat out of place, and perhaps would be better suited for *An Introduction to Microcomputers: Volume 1—Basic Concepts* (Berkeley: Osborne/McGraw-Hill, 1980).

The third chapter describes the Z8000 instruction set, and is 376 pages long. Operating modes are described, as are registers, addressing modes, and every other facet of the Z8000 instruction set. I found one peculiarity in this chapter. In discussing the Z8000 flag and control word register, this book described a bit they called the Extended Processor Architecture bit. According to this book, the bit is used to determine how extended instructions are to be processed by external devices. However, the Advanced Micro Devices literature (AMD is the second source for the Z8000) describes this bit in a different manner. The *AmZ8001/2 Processor Instruction Set* book (Advanced Micro Devices, 1979) says the bit doesn't exist at all. I suspect that the device was revised after the Osborne/McGraw-Hill book was printed, although the AMD book was corrected in 1979.

The next six chapters provide examples of the use of the Z8000. Simple programs, loops, code conversion, arithmetic problems, tables and lists, and subroutines are covered.

Chapters 11, 12 and 13 describe I/O, interrupts and memory configuration. Some discussion of the *Z8001* Memory Management Unit is included in chapter 13. Chapter 13 does a pretty good job of describing how the *Z8001* works in a large memory environment.

The remainder of the book is interesting reading for those who would like to get a picture of overall software design. Chapter 14 discusses problem definition and program design. This chapter is not tailored specifically to the Z8000, but is general in nature. Chapter 15 talks about debugging and testing software, and chapter 16 discusses documentation and redesign.

The last chapter in the book has two sample projects: a digital stopwatch and a digital thermometer (both, I'm afraid, overly simple uses of a device like the Z8000). The software design process for

the projects is given, along with hardware design and schematics, as well as complete assembly language listings.

Overall, this book covers the Z8000 pretty well. I'd give it points for readability and sheer volume. It's a big book, though well organized, and probably worth the price.

Martin Moore
Aloha, OR

Using CP/M, A Self-teaching Guide

Judi N. Fernandez and Ruth Ashley
John Wiley and Sons, Inc., 1980
$8.95

Many people's first encounter with an operating system is with Digital Research's CP/M. Unfortunately, CP/M's documentation is not written for novices. It has been described as, at best, "difficult to read" and, at worst, "a user's nightmare." To someone unfamiliar with exactly what operating systems are for, what they do, what files are and so on, CP/M's six accompanying manuals provide no place to start.

Using CP/M knows exactly where to start: at the beginning. Using a self-teaching or programmed instruction approach, this book introduces you to the simpler parts of CP/M, including all built-in commands (e.g., DIR, SAVE) and some of the commonly used transient programs (such as PIP and STAT) and includes three chapters on the CP/M editor, ED.

The writers assume almost no computer background. (There are 23 pages in the introductory section, and then you get to the section on how to type commands into the system.) If you are buying a computer system with CP/M and have never used an operating system before, or you're unclear about what operating systems do, or you just don't think you can learn about a new operating system from manuals which are not even clear as reference manuals much less as introductory texts, then you might as well get a copy of this self-teaching guide. It will certainly make things easier for you.

The book is broken into chapters, the chapters are broken into sections, and the sections are broken into frames; each frame is between a half and a whole page of text followed by questions. This is the self-teaching format. The last time I used a book of this format to learn something completely new, I found it tedious. But I also learned what I wanted to learn rather quickly. I was disappointed to see that the frame questions did not include interactive use of a machine. (One of the nice things about learning how to use a computer interactively is that it usually tells you straight out when you're wrong; no need to consult the answer section of a book.) There are suggested machine exercises at the end of each chapter.

However, if you are not a complete novice, you will be frustrated by the tone and contents of this book. It provides an introduction to only the easiest parts of CP/M. It can't go into the assembly-language interface, which is where most people really get stuck, since it can't assume the reader knows assembly language. Similarly, it can't describe the use of DDT, the interactive debugging tool. And the programmed teaching format greatly reduces its value as a reference book.

There are minor technical errors in several places. For example, CP/M does not give read-only status to new disks in drives other than drive A, as this book claims; it gives it to all disks changed without doing a warm start. One suggested machine exercise will put you into an infinite loop of interactive error messages which requires cold starting to escape. (I wonder if the authors tried their exercises.) None of the errors is especially serious, though.

The last word: If you've had your CP/M for awhile and are one of the many people frustrated with the documentation because you can't understand the nitty-gritty details of CP/M, look somewhere else. But if you're a beginner, or if you're buying your computer to use prepackaged software or to start writing BASIC

and it happens to use CP/M, then this book will make getting started easier.

Richard Fritzson
Buffalo, NY

The Micro Millennium

Dr. Christopher Evans
The Viking Press, 1980
Hardcover, 249 pp., $10.95

Dr. Evan's book came to my attention a few months ago in a *Playboy* magazine book review. The fact that *Playboy* was reviewing a book about microcomputers set an interest flag in my central processor. A month later I received an advertisement quoting Ray Bradbury as claiming that *Millennium* was one of the most mind-expanding books that he had ever read. That clinched it. I received the book on the day before a rare, short vacation. Other projects planned for the vacation remain incomplete, but I finished the book within the first two days.

The book traces the evolution of computers and their impact on culture up to the present, and projects these trends into the turn of the century. If Dr. Evans' projections are only 50 percent correct, the fireworks have just begun and the most dramatic technological advances and cultural impacts are imminent.

Millennium is divided into roughly five sections: The Past, Present, Short-Term Future (1980–1982); Mid-Term Future (1983–1990) and Long-Term Future (1991–2000). Evans contends that computers are advancing so dramatically that speculation past the year 2000 is pure recklessness.

The Past is a brief but concise description of the computer's humble beginnings, the major human stars of the story and major technological turning points. Where did IBM come from anyway? Why was the seductive Ada, Countess of Lovelace, fooling around with Babbage's machine? Dr. Evans' sense of humor is a breath of fresh air in computer literature.

The Present sets the stage for the exponential sophistication of computer technology, its impact on our culture and ultimately you and me. The material is aimed at the general reader, digressing to technical details only when absolutely necessary.

The Future sections deal with the likely areas of impact such as the death of the printed word. Why buy an expensive bulky book when you can get it on a ROM chip for 20 cents? Plug the chip into your reader and the reader will tell you the chip's contents or display it conveniently on your bedroom ceiling. At last, a truly worthwhile computer application.

Dr. Evans projects that inroads into the untouchable professions will be substantial by the mid 1980s. He argues that a professional attends school for many years to accumulate esoteric facts and relationships, then disseminates this information as needed for sometimes exorbitant fees. Surely a computer can find and dispense facts and relationships faster, more accurately, and probably a lot less expensively than a doctor, lawyer or Indian chief.

After all the social upheaval caused by the fundamentalist anticomputer groups, the declining number of hours in the work week and the decline of the work ethic in general, what will we do with our spare time? You guessed it. Turn computer power onto the long-neglected problems of education, medicine, science, man and his relation to the universe.

What kind of computer tools might we expect to have at the turn of the century? Fifteen years ago experts said no computer could be programmed to play a decent game of chess; today there are computers which can beat 99.5 percent of the world's chess players. Soon they will be at the International Master level. How much longer will it be before the chess champion of the world is a computer? Now things really get interesting. If man can design a machine that can consistently out-think him in some area, can he not design an ultra-intelligent machine to design the next generation of UIMs to push the man-machine intelligence gap even further? Most of us that have had

much contact with computers have briefly considered the can-computers-think question, and answered smugly in the negative. If you read this book you may want to reevaluate your position.

The Micro Millennium is a thought-provoking glimpse into the near future. If the computer revolution catches you unprepared, don't say Dr. Evans didn't warn you.

J. A. Cuchiara
Encinitas, CA

For Me and You and the 1802

Jeff Duntemann
301 Susquehanna Road
Rochester, NY 14618
Paperback, 30 pp., 1980

Finally, a book about the Cosmac 1802! This often underrated microprocessor has given many hobbyists a start in microcomputing, through the home-built Cosmac Elf and its commercial cousins. *Captain Cosmo's Whizbang* is all about such small 1802 systems, from the author's adventures in building one to plans for expanding the trainer to a full-blown microcomputer system.

The book consists of 30 pages (large pages and small type) of articles assembled in loose order. A comprehensive index, rather than a table of contents, is provided. In these dozens of articles, from a paragraph to more than a page long, is a wealth of information. There are articles on building and modifying an Elf, parts, wire-wrapping, programming, expansion, music, video, mass storage and power supplies, to name a few of the topics covered. There are program listings for sound effects, animation, Morse code, video sketchpad and more, ready to key in. Throughout the book you will be amused by the antics of Cosmo the robot, who advises: "MOS can often be found on the north side of trees!"

As a veteran Elf hacker, I found Duntemann's book highly informative. Articles such as the one describing the VIP bus structure have inspired projects I will eventually implement on my own Elf. A beginner would not feel lost in this sea of information, either, for much of it is oriented toward the kind of novice the author once was, as is shown by the introduction:

". . . I'll never forget the incandescent blaze of understanding that roared out of my ears when another computer freak friend finally explained what an instruction set was . . . EPIPHANY!!!! A COMPUTER IS A BOX WHICH FOLLOWS A PLAN! . . . With that out of the way, I knew I had to have one."

One of the book's best features—too seldom seen in computer literature—is Duntemann's straightforward, conversational style. His articles do not have to be read over and over just to glean the bare facts; he presents the most complex subjects in a clear, understandable form.

The only criticism I have is that the programs presented are mostly for systems, like the VIP, with more than 256 bytes of RAM. Perhaps this is just a gripe from someone who has not expanded the original single-page Elf memory yet. Anyway, he does include a simple and inexpensive plan for memory expansion, perhaps with such situations in mind.

Captain Cosmo's Whizbang is a must for anyone who owns an 1802 system. Even Studio II owners can find out how to harness the computer hiding inside their TV game. For those who are just thinking about building or purchasing an 1802 system, the book is a valuable source of information and ideas; the author discusses the Cosmac's flaws as well as its advantages.

Most of you reading this review have never even heard of the book before, as it has not been widely advertised. Word of mouth will provide most of the publicity it gets, so tell all your 1802-hobbyist friends. When you get *Captain Cosmo's Whizbang* you won't be able to wait to try out all of the projects and programs.

Larry Stone
New Haven, CT

Call For Manuscripts

Kilobaud Microcomputing is looking for business articles!

Businessmen in all fields are beginning to take notice of the microcomputer. They are eager to know which computers, peripheral equipment and applications software will let them take full advantage of this new tool. What knowledge do you have to share?

Here are the kinds of articles that we want you to write for us:

- Are you a businessman with a system up and running? We want to know how it works. What were your expectations? Have they been fulfilled? Did you find the software that you wanted? What problems have you had? How did you overcome them? What recommendations do you have for other businessmen?
- We want reviews from a businessman's perspective of specific hardware and software. If you've recently bought a new product and want to tell others how great—or poor—it is, *Microcomputing* will provide you with a forum.
- What programs have you written to meet your specific needs? Perhaps another businessman can use them, too. Even if he can't, your program may serve as a springboard for other ideas.
- Perhaps you aren't using your micro for business, but know a company that is. Trot on down with your pencil and notebook, and find out what they're up to. While they might not have the time to write up their experiences, they might be more than willing to tell somebody else about them. And an outside observer will often be able to see things with a unique and valuable perspective.

Don't worry if you're not a professional writer. That's what we editors are here for. And we'll be more than happy to send you a copy of our writer's guidelines.

Send your manuscripts and correspondence to:
Kilobaud Microcomputing
Pine St.
Peterborough, NH 03458

PERSPECTIVES

(from page 242)

business or plant because of the heavy costs of the equipment and staff it required. Often, only the heavier accounting functions have had the heaviest applications, and other functions, such as purchasing, continue as before.

Most of the challenge to reduce clerical work has not yet been met. In any medium-sized business you will see inordinate amounts of clerical work still being done. These are the smaller jobs which everyone does.

A Call to the Industry

In general, the microcomputer industry appears to have done everything except look at the user and the application. That is not all bad. The equipment is impressive.

So my call goes out to the industry. Give us microcomputers for micro applications, and we will beat a path to your door. All those applications will make for macro demand, once you address the need.

While you are at it, give us Pascal in ROM. Give us the best tools you have for us.

Look at what has gone before, the still ubiquitous adding machines and calculators, and give us what we need to cut down those millions of unnecessary keystrokes and pencilings, and print the results of a series of computations in a programmable "report" format so that no further work is required.

Who will be the first to offer the equipment we need?□

Terminals and Printers

Intertec
EMULATOR

Software compatible with a Soroc IQ-120, Hazeltine 1500, ADM-3A or DEC VT-52. Features block mode transmission and printer port; 12" anti-glare screen; 18-key numeric keypad; full cursor control. List $895

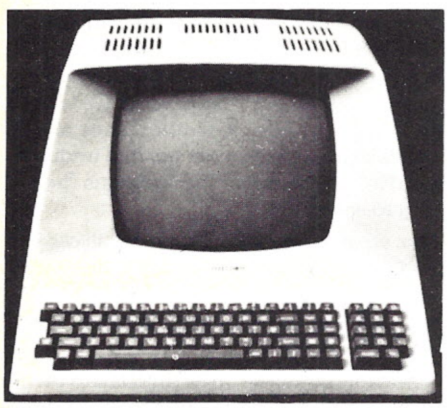
OUR PRICE $749

NEW INTERTUBE III

List $995 **ONLY $749**

12" display, 24 x 80 format, 18-key numeric keypad, 128 upper/lower case ASCII characters. Reverse video, blinking, complete cursor addressing and control. Special user-defined control function keys, protected and unprotected fields. Line insert/delete and character insert/delete editing, eleven special line drawing symbols.

TELEVIDEO TVI-912C



Upper and lower case, 15 baud rates: 75 to 19,000 baud, dual intensity, 24 x 80 character display, 12 x 10 resolution. Numeric pad. Programmable reversible video, auxiliary port, self-test mode, protect mode, block mode, tabbing, addressable cursor. Microprocessor controlled, programmable underline, line and character insert/delete. "C" version features typewriter-style keyboard. List $950

OUR PRICE $789

920C (with 11 function keys, 6 edit keys and 2 transmission mode keys, List $1030

ONLY $849

950C List $1195 . . . . $1039

HAZELTINE

1500

ONLY

$879

1410 w/numeric keypad, List $900 . . . . . . **$749**
1420 w/lower case and numeric pad **849**
1510, List $1395 . . . . **1089**
1520, List $1650 . . . . **1389**

SOROC

IQ-120

List $995

SPECIAL

$729

IQ135
$749

IQ-140 List $1495
SPECIAL $1149

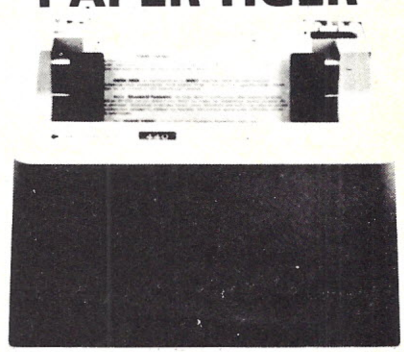
CENTRONICS
PRINTERS

NEW 730, parallel, friction, tractor $599
NEW 737 parallel, friction, tractor $799
779-2 w/tractor (same as TRS-80 Line Printer I), List $1350 . . . . $799
704 RS232 serial version, $2350 . . . . $1595
704-9 RS232 180CPS $1595
704-11 Parallel 180CPS $1695

NEC SPINWRITER™

Terminal/Keyboard as well as RO Printer Only models available.
CALL FOR PRICES!

PAPER TIGER®

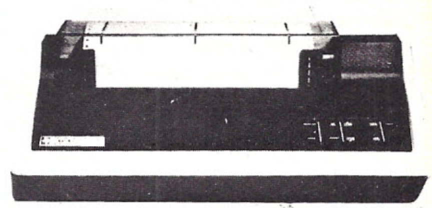


IDS 445G PAPER TIGER $698
Buffer w/graphics option, incl. buffer, . . $789
TRS-80 cable $45

NEW IDS 460
QUALITY PRINTING AT MATRIX SPEED—LOGIC SEEKING PROPORTIONAL SPACING
w/auto text justification

NEW IDS PAPERTIGER 460 List $1295 . **$1099**
NEW IDS PAPERTIGER 460G List $1394 **$1149**
NEW IDS 560G List $1794 . . . . $1499

TI-810



TI-810 Basic Unit, $1895 . **ONLY $1595**
TI-810 w/full ASCII (Lower case), vertical forms control, and compressed print . **$1795**
TI-745 Complete printing terminal with acoustic coupler, List $1695 . . . . **$1399**

ANADEX
DP9500/DP9501 PRINTERS

DP-9500, List $1650 . . . . **$1349**
DP-9501, List $1650 . . . . **$1349**

ANADEX DP-8000
80-Col. Dot Matrix. **$849**

OKIDATA
Microline 80 List $599 **ONLY $499**
Tractor Feed Option **$109**
Serial interface . . . . **$ 99**
Microline 82 List $799 . . . . $679
Microline 83 List $1260 . . . . $1069

AXIOM IMP I $699

Epson MX-80 List $645 . . . . $499

Above prices reflect a 2% cash discount (order prepaid prior to shipment). Add 2% to prices for credit card orders, C.O.D.'s, etc. Prices are f.o.b. shipping point. Prices are subject to change and offers subject to withdrawal without notice. **WRITE FOR FREE CATALOG.**

MiniMicroMart, Inc.

✓226

VISA

master charge

1618 James Street, Syracuse NY 13203 (315) 422-4467 TWX 710-541-0431

By Martin Klaver

White-Collar Micro Worker

Industry Must Meet His Demands

To an industrial engineer like myself (really an efficiency engineer), whose main purpose is to cut costs for a business or manufacturing plant, the microcomputer is the most exciting, and yet the most frustrating, piece of equipment to come down the road in years.

We industrial engineers must find out where the costs are highest and get whatever is needed to cut those costs, whether it is a new system such as an incentive, or a new piece of equipment such as a microcomputer. As a project engineer also responsible for planning, I need to look up ahead, get ready for the future and make that future happen.

I can see an explosion coming in the use of microcomputers. I wonder how long it will take for equipment manufacturers to get over the last hump to widespread use in medium-sized businesses and manufacturing plants, where there is a crying need to streamline clerical work and increase industrial productivity.

Let's take a quick look at what's good and bad about what is already on the market.

First, the price of equipment is already down there where it needs to be, and that is the hardest hurdle to jump.

Second, the equipment has the memory and program capabilities needed to handle most applications.

Third, BASIC provides a fairly good method of programming. And here is the first qualification—"fairly good." The microcomputer industry has a better language, Pascal, which makes better use of the microcomputer and the programmer, and any edge in efficiency cannot be overlooked by an industrial engineer. Still, BASIC will certainly do for now.

Martin Klaver is Senior Project Engineer for Special Projects and Planning at Lenox China, Inc., Pomona, NJ.

So then, where are the problems?

Is it in marketing, which appears to target the hobby or game interest first? Is that scaring off serious applications. I doubt it.

Despite all the emphasis given to the "game" image, any industrial engineer worth his salt would use any equipment needed to cut costs, no matter what it was promoted for. I have seen roller skate wheels put to work in the design of a machine.

Can it be that industrial engineers are unfamiliar with the possibilities of microcomputers to cut down clerical work? That may be true in part, but you will see industrial engineers using microcomputers to control nearly all small machines in production operations well before desk applications are fully exploited. The machine applications will come before the white-collar applications for the following reasons.

1. The machine hardware is there now, in the proper configuration.
2. EPROM controls are cheaper than conventional controls in many cases.
3. EPROM controls will cut maintenance costs and improve reliability.
4. Microcomputers, using a display, can actually tell the machine operator what to do and when to do it. This enforces operating procedures like never before; unless procedure is followed, the machine does not operate.

In fact, if you look closely at the new machine controls, which come in great variety, they have some of the attributes microcomputers will need if they are to be successfully used for reduction of clerical work.

1. They normally have a numeric keypad, which is absolutely essential in an office machine, too.
2. They can tell the operator what to do through prompting.
3. They can be programmed in one place, using a microcomputer to produce a master program that can be run elsewhere by slave units.

Desk machines, intended for clerical work, also need keypads, prompting and master-slave configuration to be effective in reducing clerical work done over and over again, whether it be a short-term application done eight hours a day until that particular job is done or a regular job done a few hours a day every day or every few days. The person doing the clerical job may not be a clerk; indeed, everybody deals with figures.

We industrial engineers need units we can quickly program for specific and specialized applications and hand to someone in a portable package to run that program, with full prompting. We can save everybody's time with such applications, including the time of the most highly paid people in our companies. We can give them the use of mathematics and systems they do not even need to understand in order to solve the problems they need to solve.

So, why not use the equipment already available?

When you start seriously looking, you find that the right combination is not there.

The micros appear to be directed to the macro applications, such as payroll, which in most medium-sized businesses are already handled by central data processing systems. These larger systems may have programming which is obsolete and costly by present-day standards, but they will not be replaced by microcomputers.

The larger data processing systems took the icing off the cake, but most of the cake is still there, in the form of micro applications, the smaller jobs which make up most of the work.

It was never worthwhile for large data processing systems to handle most of the clerical work done in a medium-sized

(continued on page 238)

A cost effective 10 megabyte system from the leader in Winchester based microcomputers.

Ohio Scientific has put a low-cost, high-performance 8″ non-removable hard disk together with its popular desk-top microcomputer system. This yields approximately 10 megabytes of fast hard disk capability at a tremendous cost/performance benefit over floppy based microcomputer systems.

C3-D

The 10 megabyte system is also available with the added advantage of triple microprocessors — the 6502A, 68BOO and Z-80A. This allows you to make maximum use of Ohio Scientific's extensive software library as well as programs offered by independent suppliers and publishers. $7,600.

C2-D
Standard Features

- 52K RAM
- 8″ floppy disk drive for program transport and backup.
- OS-65U small business operating system.
- 9-digit precision BASIC by Microsoft.
- Available in OEM quantities at attractive discounts.

For literature and the name of your local dealer, CALL 1-800-321-6850 TOLL FREE.

OHIO SCIENTIFIC
a M/A-COM Company
1333 SOUTH CHILLICOTHE ROAD, AURORA, OH 44202 • (216) 831-5600